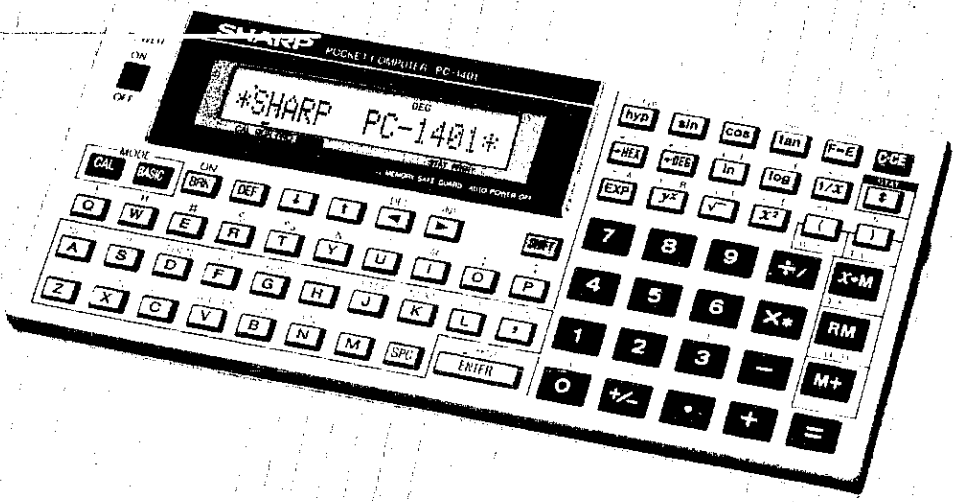
SHARP

TASCHENCOMPUTER

MODELL PC-1401 PC-1402

BEDIENUNGSANLEITUNG



SHARP PC - 1401/1402

BEDIENUNGSANLEITUNG

Lieber Leser !

Wir haben uns bemüht, dieses Handbuch ohne Fehler zu erstellen. Doch auch bei der sorgfältigsten Prüfung kann noch etwas übersehen werden. Wenn Sie daher Fehler finden und/oder den einen oder anderen Verbesserungsvorschlag zu diesem Handbuch haben, so teilen Sie uns dies bitte mit.

Vielen Dank im voraus

SHARP ELECTRONICS
HAMBURG

HQE 5/86

EINLEITUNG
I Allgemeines 7
II Betriebshinweise. 8
III Stromversorgung 9
IV Tastaturabdeckung 11

TEIL I - THEORIE

1.1 GRUNDLAGEN SHARP PC-1401/1402 15

1.1.1 Einschalten des Computers. 15
1.1.2 Ausschalten des Computers. 15
1.1.2.1 Automatisch. 15
1.1.2.2 Manuell. 15
1.1.3 Bedienelemente 15
1.1.4 Tastenfunktionen 16
1.1.5 Die Anzeige. 17
1.1.6 Wahl der Betriebsart 20
1.1.6.1 CAL-Mode 21
1.1.6.2 RUN- und PRO-Mode. 22
23

1.2 GRUNDLAGEN SHARP CE-126P 24

1.2.1 Allgemeines. 24
1.2.2 Verwendung als Drucker 24
1.2.2.1 Manueller Druckbetrieb (Protokollerstellung) 25
1.2.2.2 Programmgesteuerter Ausdruck 25
1.2.3 Verwendung des integrierten Kassetten-Interfaces . . . 26
1.2.3.1 Auswahl des Kassettenrekorders 27
1.2.3.2 Anschluß des Kassettenrekorders an das CE-126P . . . 27
1.2.4 Datenspeicherung auf Magnetband. 28
1.2.4.1 Allgemeines. 28
1.2.4.2 Speichern von Daten und Programmen 28
1.2.4.3 Überprüfen der Abspeicherung 29
1.2.4.4 Laden von Programmen oder Daten. 30

TEIL II - PRAXIS

2.1 SHARP PC-1401/1402 - EINSATZ ALS WISSENSCHAFTLICHER RECHNER 33

2.1.1 Betriebshinweise 33
2.1.1.1 Zweite Funktionsebene der Tastatur; Merkmale; Editier- und Korrekturmöglichkeiten. 33
2.1.1.2 Anzeige/Display-Format und Symbole 40
2.1.1.3 Grundeinstellung 41
2.1.1.4 Löschen der Eingabe oder einer Fehlermeldung 42

2.1.1.5 Auf-/Abrundung bzw. Festlegung der Nachkommastellen . 43
2.1.1.6 Wissenschaftliche Schreibweise (Notation) 44
2.1.1.7 Umwandlung des Winkels und der Zeit 45
2.1.1.8 Koordinatenumwandlung 46
2.1.1.9 Umwandlung "Hexadezimal - Dezimal" und Rechnen mit Zahlen in Hexadezimal-Schreibweise. 46
2.1.2 Normale Berechnungen. 49
2.1.2.1 Addition und Subtraktion. 50
2.1.2.2 Multiplikation und Division 50
2.1.2.3 Potenz- und Wurzelfunktion. 51
2.1.2.4 Prozentrechnung 51
2.1.2.5 Reziprok-Rechnung 52
2.1.2.6 Speicherrechnung. 52
2.1.2.7 Vorrangordnung und Verwendung der "Klammern"-Tasten . 53
2.1.3 Wissenschaftliche Berechnungen. 55
2.1.3.1 Trigonometrische und inverse trigonometrische Funktionen (Arcusfunktionen). 55
2.1.3.2 Hyperbel- und inverse Hyperbelfunktionen (Area-funktionen) 56
2.1.3.3 Logarithmische Funktionen 56
2.1.3.4 Exponentialfunktionen 57
2.1.3.5 Fakultät. 57
2.1.4 Statistische Berechnungen 57
2.1.5 Rechenbereich 64

2.2 SHARP PC-1401/1402 - EINSATZ ALS BASIC-RECHNER 68

2.2.1 Rechnen ohne Programmunterstützung (RUN-Mode) 68
2.2.1.1 Grundrechnungsarten 68
2.2.1.2 Rechengenauigkeit 70
2.2.1.3 Wissenschaftliche Schreibweise (Notation) 71
2.2.1.4 Editier-/Korrekturmöglichkeit 72
2.2.1.5 Mathematische Funktionen/Klammerregeln. 73
2.2.1.6 Hexadezimal-(Sedezimal-)Zahlen. 74
2.2.1.7 Textausdrucke 75
2.2.1.8 Logische Vergleichsausdrücke. 76
2.2.2 Sprachelemente. 78
2.2.2.1 Numerische Konstante. 78
2.2.2.2 Textkonstante 78
2.2.2.3 Nummerische Variable. 78
2.2.2.4 Textvariable. 79
2.2.2.5 Numerische Funktionen, Textfunktionen 79
2.2.2.6 Numerischer Ausdruck. 80
2.2.2.7 Textausdruck. 80
2.2.2.8 Logische Vergleichsausdrücke. 81
2.2.3 Variablen im Sharp PC-1401/1402 83
2.2.3.1 Standartvariable. 83
2.2.3.2 Einfache Variable 85
2.2.3.3 Indizierte Variable 85
2.2.3.4 Besonderheiten der Variablen A. 86

2.2.3.5 Feldvariablen 87
2.2.3.6 Numerische Feldvariablen. 88
2.2.3.7 Textfeldvariablen 89
2.2.4 Arithmetische Funktionen. 91
2.2.5 Textfunktionen. 92
2.2.6 Programmieren in BASIC. 93
2.2.6.1 BASIC-Obersicht 93
2.2.6.2 Programmerstellung. 94
2.2.6.3 Programmaufbau. 95
2.2.6.4 Eingabe eines Programms 95
2.2.6.5 Korrektur einer Zeile 97
2.2.6.6 Löschen einer Zeile 98
2.2.6.7 Programmausführung. 98
2.2.6.8 Fehlermeldungen/Fehlersuche 99
2.2.7 PC-1401/1402-BASIC - Befehlsvorrat. 100
2.2.7.1 Fest vorprogrammierte Schlüsselwörter und arithmetische Funktionen 100

ANHANG

A. TASTENFUNKTIONEN DES PC-1401/1402. 212
B. LISTE DER FUNKTIONEN UND ANWEISUNGEN 214
C. BASIC - BEFEHLSOBERSICHT (Kurzfassung) 215
D. STANDARDVARIABLEN IM PC-1401/1402 225
E. OPERATOREN 226
E.1 Arithmetische Operatoren 226
E.2 Textfunktionsoperatoren 226
E.3 Logische Vergleichsoperatoren 226
E.4 Reihenfolge der Operatoren 226
F. RECHENBEREICH 226
G. ABGELEITETE MATHEMATISCHE FUNKTIONEN 227
H. PROGRAMM-HAUPTSPEICHER-BEGRENZUNG 229
I. FEHLERMELDUNGEN 230
J. ASCII-CODE TABELLE 231
K. REFERENZLISTE 232
L. TECHNISCHE DATEN PC-1401/1402 233
M. TECHNISCHE DATEN CE-126P 234
N. INDEX 235

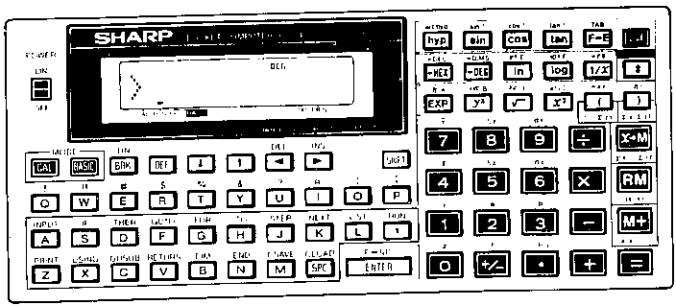
E I N L E I T U N G

E I N L E I T U N G

I ALLGEMEINES

Mit dem **PC-1401/1402** stellt SHARP einen preisgünstigen und dennoch außerordentlich leistungsstarken BASIC-Taschencomputer mit integriertem wissenschaftlichem Rechner vor.

Erweitertes BASIC, 40k-ROM-Betriebssystem und 4,2k-RAM-Bereich (PC-1401), 10,2k-RAM-Bereich (PC-1402) für Anwenderprogramme sowie 48 vorprogrammierte mathematisch wissenschaftliche Funktionen lassen diesen Rechner auch gehobenen Ansprüchen aus fast allen Bereichen gerecht werden.



Frontansicht des **PC-1401/1402**

Zusammen mit dem Thermodrucker mit integriertem Kassetten-Interface **CE-126P** wird der **PC-1401/1402** zu einer kleinen, leistungsfähigen Datenverarbeitungsanlage.

Bei der Gestaltung dieser Bedienungsanleitung wurde versucht, sowohl den Ansprüchen eines Anfängers als auch denen eines geübten Programmierers gerecht zu werden. In manchen Fällen war eine Kompromißlösung aber nicht zu umgehen, und wir verweisen daher bereits an dieser Stelle auf die umfangreiche BASIC-Literatur, die im Fachhandel erhältlich ist. Eine Referenzliste mit genauen Bestellangaben finden Sie im Anhang dieses Handbuchs.

Wir hoffen, Ihnen mit diesem Handbuch alle möglichen Hilfsmittel in die Hand zu geben, so daß Sie Ihren **PC-1401/1402** für Ihre Belange optimal einsetzen können.

II BETRIEBSHINWEISE

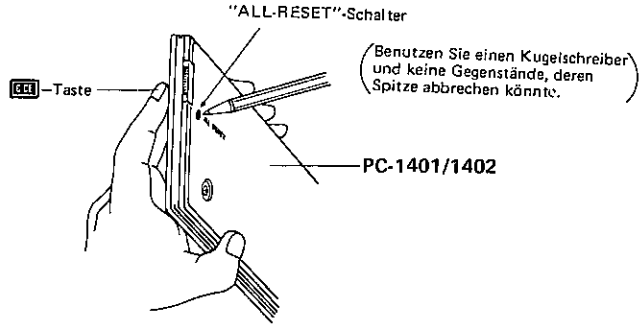
1. Die Flüssigkeitskristallanzeige des **PC-1401/1402** ist in Spezialglas eingebettet und kann daher bei Gewalteinwirkung zerbrechen. Behandeln Sie den Computer deshalb mit Sorgfalt. Beim Transport immer die Tastaturabdeckung verwenden.

2. Schützen Sie den Computer vor Staub, Feuchtigkeit und allzu großen Temperaturschwankungen.

3. Zur Reinigung dient ein weiches, trockenes Tuch. Keine Reinigungs- oder Lösungsmittel verwenden.

4. Durch elektrostatische Entladungen über den Computer oder durch Fehlbedienung (der Computer blieb beim Batteriewechsel oder Anschluß der Option **CE-126P** eingeschaltet) kann der **PC-1401/1402** "abstürzen". Dadurch werden alle Tastenfunktionen einschließlich der [CAL] -Taste blockiert.
   Sollte Ihnen dies passieren, müssen Sie den Computer **PC-1401/1402** initialisieren. Dazu stehen zwei Möglichkeiten zur Auswahl:

   1) Drücken Sie die [CAL] -Taste und betätigen Sie den "ALL RESET"-Schalter an der Rückseite des Geräts. Bei dieser Art der Initialisierung werden die Programme, Variablen und der Reservespeicher nicht gelöscht.



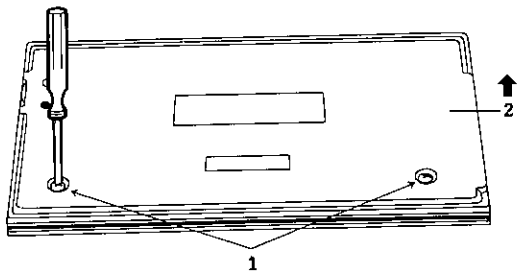
Initialisieren des **PC-1401/1402**

   2) Sollte dadurch die Blockade immer noch nicht aufgehoben sein, betätigen Sie den "ALL-RESET"-Schalter, ohne dabei die [CAL] -Taste zu drücken. Alle Speicherinhalte werden dadurch gelöscht.

III. STROMVERSORGUNG

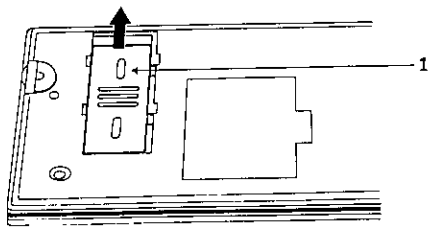
Der Taschencomputer **PC-1401/1402** wird mit Lithiumbatterien betrieben. Die Batterien sind werksseitig bereits eingesetzt. Bei erschöpften Batterien muß ein Wechsel wie folgt vorgenommen werden:

1. Den Computer ausschalten.
2. Die zwei Gehäuseschrauben (1) an der Rückseite des Geräts entfernen.
3. Den Gehäusedeckel (2) auf der Schraubenseite etwas anheben und in Richtung des Pfeils schieben und abheben.



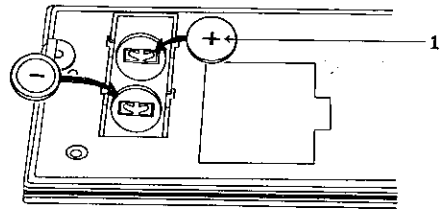
Entfernen des Gehäusedeckels

4. Die Batterieabdeckung (1) in Richtung des Pfeils schieben und dem Rechner entnehmen.



Abnehmen der Batterieabdeckung

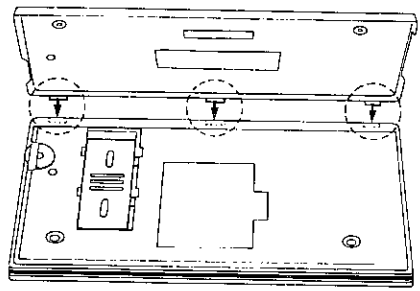
5. Die erschöpften Batterien (1) durch neue ersetzen.
   (Batterien nur paarweise wechseln)



Batteriewechsel

6. Die Batterieabdeckung wieder einsetzen und verriegeln.

7. Den Gehäusedeckel wie dargestellt auf den Rechner aufsetzen und verschrauben.



Aufsetzen des Gehäusedeckels

Folgende Punkte sind beim Batteriewechsel besonders zu beachten:

- Wechseln Sie die Batterien immer paarweise.
- Achten Sie darauf, daß Sie beim Batteriewechsel die alten Batterien nicht mit den neuen vertauschen.
- Verwenden Sie nur den angegebenen Batterietyp.
  Batterietyp: CR-2032 oder Ersatztype

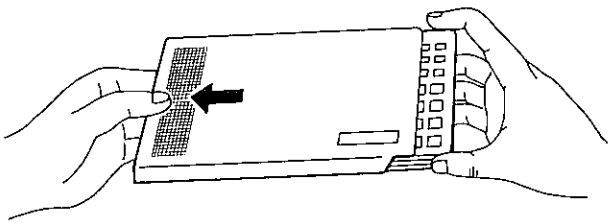
Die Batterien von Kindern fernhalten.

IV. TASTATURABDECKUNG

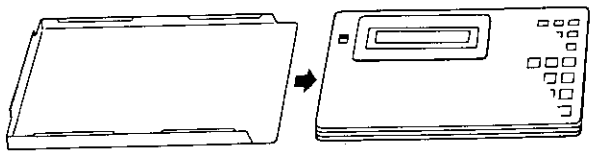
Zum Schutz der Tastatur und der Anzeige ist der **PC-1401/1402** mit einer stabilen, doppelseitig aufschiebbaren Tastaturabdeckung ausgestattet.

Die Tastaturabdeckung des **PC-1401/1402** wie dargestellt verwenden.

Bei Gebrauch:

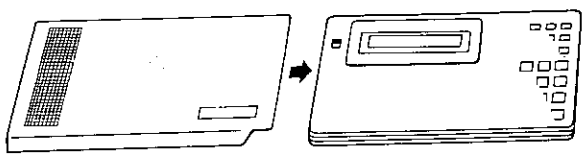


Die Tastaturabdeckung ist so gestaltet, daß sie während der Verwendung des Rechners auf dessen Unterseite geschoben werden kann.



Tastaturabdeckung bei Gebrauch des Rechners

Nach Gebrauch:



Tastaturabdeckung nach Gebrauch des Rechners

T E I L I

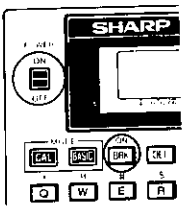
T H E O R I E

T E I L I T H E O R I E

1.1 GRUNDLAGEN PC-1401/1402

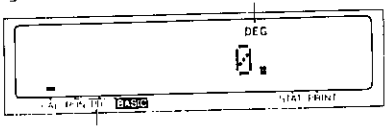
1.1.1 EINSCHALTEN DES COMPUTERS

Das Einschalten des **PC-1401/1402** erfolgt zum einen über den Schiebeschalter links neben dem Display, zum anderen über die Taste ON BRK .



Nach dem Einschalten erscheinen auf der Anzeige folgende Symbole:

Zeigt die Winkeleinheit (DEG/GRAD/RAD) an.



Zeigt Betriebsart (CAL/RUN/PRO) des **PC-1401/1402** an.

1.1.2 AUSSCHALTEN DES COMPUTERS

1.1.2.1 Automatisch

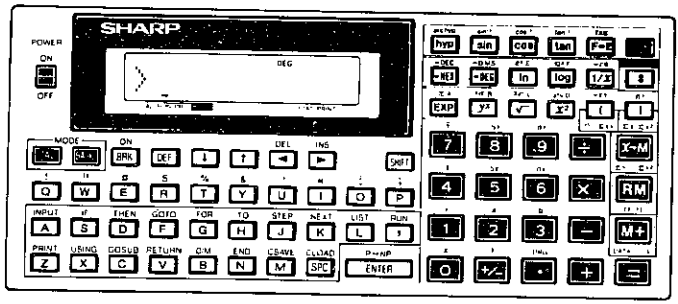
Der Computer schaltet sich nach ca. 11 Minuten nach der letzten Tastenbedienung automatisch ab. Dabei bleiben alle Programm- und Dateninformationen gespeichert.

Hat sich der Rechner automatisch ausgeschaltet, muß er über die Taste ON BRK wieder aktiviert werden. Auf der Anzeige erscheint die zuletzt bearbeitete oder angezeigte Zeile.

1.1.2.2 Manuell

Soll der Computer sofort nach Gebrauch ausgeschaltet werden, so muß der Schiebeschalter links neben dem Display in Stellung OFF gebracht werden. Die Programm- und Dateninformationen bleiben dabei gespeichert. Der Computer läßt sich auch während einer Programmausführung abschalten.

1.1.3 BEDIENELEMENTE



Das Bedienfeld des **PC-1401/1402** besteht aus 76 Tasten, die zu 4 Blöcken zusammengefaßt sind:

- die Schreibmaschinen-Alphatastatur mit [ENTER]-Taste (2 Ebenen)
- die Tasten für die Sonderfunktionen
- die numerische Tastatur mit den Grundrechnungsarten (2 Ebenen)
- die Tasten für die wissenschaftlichen Funktionen (2 bzw. 3 Ebenen)

Die Schreibmaschinen-Alphatastatur ist mit zwei Ebenen belegt. Bei einfachem Tastendruck gelten die auf den Tasten stehenden Bezeichnungen. Will man die über den Tasten stehenden Sonderzeichen (oberste Reihe der Tastatur) oder die vorprogrammierten BASIC-Anweisungen abrufen, muß vorher die Doppelfunktionstaste [SHIFT] gedrückt werden.

Mit der Taste [DEF] kann den unteren beiden Reihen noch eine dritte Ebene zugewiesen werden. Über die sogenannten 'Definable Keys' können verschiedene Programme sofort gestartet werden; man drückt dazu die Taste [DEF] und eine dem Programm zugewiesene Taste.

In der Reihe über der Alpha-Tastatur befinden sich die Sonderfunktionstasten zur Wahl der Betriebsart, zum Unterbrechen oder zum Auflisten sowie zum Editieren eines Programms.

Rechts neben der Alpha-Tastatur befindet sich die numerische Tastatur. Die zweite Ebene der numerischen Tastatur, die hauptsächlich für statistische Berechnungen dient, ist ebenfalls mit der [SHIFT]-Taste erreichbar.

Mit den Tasten oberhalb der numerischen Tastatur lassen sich die meisten wissenschaftlichen Berechnungen, wie trigonometrische Berechnungen, Hyperbelberechnungen, Polarkoordinatenberechnungen usw., durchführen. Dieser Teil der Tastatur ist mit bis zu drei Ebenen belegt.

Die [SHIFT]-, [BASE] - und [TYPE]-Tasten sind Wechselschalter, d.h. durch einmaliges Drücken wird in den jeweiligen Mode umgeschaltet, durch nochmaliges Drücken der Urzustand wieder hergestellt.

1.1.4 TASTENFUNKTIONEN

[DEF]	In Verbindung mit den Tasten A, S, D, F, G, H, J, K, L, ' Z, X, C, V, B, N, M, SPC ermöglicht diese Taste das Starten des Programms über einen Markennamen.
[SHIFT]	Doppelfunktionstaste
[↓] [↑]	Editiertasten
[◀]	Cursor nach links; Editieraufruf
[▶]	Cursor nach rechts; Editieraufruf
[BRK]	Unterbrechen des Programmablaufs. Am Display erscheint die Meldung BREAK IN XXX (XXX = Zeilennummer) unterbricht auch Drucker- und Cassettenbetrieb.
[CLS]	Löscht die Anzeige und die Fehlermeldungen.
[SHIFT] [▶] INS	Einfügen einzelner Zeichen (Platzhalter ␣ wird gesetzt)
[SHIFT] [◀] DEL	Löschen einzelner Zeichen
ON [BRK]	Einschalten des Rechners nach automatischer Abschaltung
CA [CLS]	Löschen der Anzeige und - Löschen des WAIT-Intervalls - Löschen des USING-Formats - Aufheben des TRACE-Betriebs - Löschen der Fehlermeldungen
[A]-[Z]	Alphatastatur A-Z; Variablennamen
[SPC]	Leerzeichen
[ENTER]	Beschließt die Eingabe einer Programmzeile. Beim Rechnen ohne Programmunterstützung im RUN-Modus ersetzt die Taste das "="-Zeichen. Programmstart in Verbindung mit RUN oder GOTO.

[!] ~ [&]	Sonderzeichen
[SHIFT] [A] ~ [SHIFT] [SPC]	Aufruf der wichtigsten BASIC-Anweisungen
[SHIFT] [ENTER] (P ↔ NP)	Umschaltung für Druck/Nichtdruck beim Rechnen ohne Programmunterstützung im RUN/PRO-Modus
[.]	Dezimalpunkt
[0] ~ [9]	numerische Zeichen
[+] ~ [÷]	Grundrechnungsarten
[hyp]	Umschaltung Hyperbelfunktionen
[SHIFT] [arc hyp]	Umschaltung Umkehrfunktionen der Hyperbelfunktionen
[sin]	Trigonometrische Funktionen
[SHIFT] [sin⁻¹]	Umkehrfunktionen der Trigonometrischen Funktionen
[F↔E]	Umschaltung Fließkomma/Festkomma
[SHIFT] [TAB]	Festlegung der Nachkommastellen
[→HEX]	Umrechnung Dezimalsystem - Hexadezimalsystem
[SHIFT] [→DEC]	Umrechnung Hexadezimalsystem - Dezimalsystem
[→DEG]	Umrechnung Dezimalsystem - Sexagesimalsystem
[SHIFT] [→D.MS]	Umrechnung Sexagesimalsystem - Dezimalsystem
[ln]	Natürlicher Logarithmus
[SHIFT] [e^x]	Exponentialfunktion e^x
[log]	Zehnerlogarithmus
[SHIFT] [10^x]	Exponentialfunktion 10^x
[1/x]	Reziprokwert
[SHIFT] [→XY]	Umwandlung Polarkoordinaten in rechtwinklige Koordinaten
[SHIFT] [→rθ]	Umwandlung rechtwinklige Koordinaten in Polarkoordinaten
[EXP]	Exponentialdarstellung
[SHIFT] [π]	Konstante PI

[SHIFT] [^]	Potenzierfunktion
[SHIFT] [$\sqrt[x]{y}$]	X-te Wurzel
[$\sqrt{}$]	Quadratwurzelfunktion
[SHIFT] [$\sqrt[3]{}$]	Dritte Wurzel
[x^2]	Quadratzahlfunktion
[SHIFT] [n!]	Fakultät
[(] [)]	Klammern
[X-M] – [M+]	Speicherrechnung
[SHIFT] [DRG •]	Umschaltung der Winkeleinheit
[SHIFT] [>]	Logische Vergleichsoperatoren
[SHIFT] [<]	
[SHIFT] [STAT ↕]	Statistikbetriebsart

1.1.5 DIE ANZEIGE

Der **PC-1401/1402** verfügt über eine 16stellige alphanumerische Flüssigkristallanzeige. Die einzelnen Zeichen werden in einer 5 x 7-Punktmatrix dargestellt.

Das Eingaberegister faßt 80 Zeichen. Alle Zeichen, die über die 16 möglichen darstellbaren Zeichen der Anzeige hinausgehen, müssen mit den Cursor-Tasten (▷ und ◁) 'gescrolled' werden.

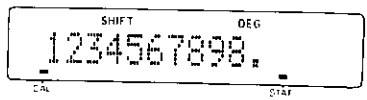
Der Cursor zeigt die Stelle an, an der die nächste Eingabe erfolgen kann. Er steht normalerweise auf der ersten freien Stelle und wird durch einen kurzen waagrechten Strich (_) dargestellt.

Eine Ausnahme besteht, wenn auf der Anzeige links das Bereitschaftssymbol (>) angezeigt wird. Durch das nächste einzugebende Zeichen wird dieses überschrieben. Erst dann sieht man den Cursor auf der ersten freien Stelle nach dem zuletzt eingegebenen Zeichen.

Stellt man den Cursor mit Hilfe der ▷- oder der ◁-Taste über ein bereits eingegebenes Zeichen, so erkennt man die Position am Blinken des Cursors an dieser Stelle in der vollen Punktmatrix.

Anhaltender Druck auf die Cursor-Tasten läßt den Cursor schnell an die gewünschte Stelle der Zeile bringen. Ein kurzer Druck verschiebt die Anzeige nur um ein Zeichen.

Versucht man mehr als 80 Zeichen einzulesen, werden diese vom Rechner nicht mehr angenommen. Jede zusätzliche Eingabe überschreibt das zuletzt eingegebene Zeichen. Der Cursor verändert hier sein Aussehen; er blinkt auf dem letzten Zeichen in der vollen Punktmatrix.



1.1.6 WAHL DER BETRIEBSART

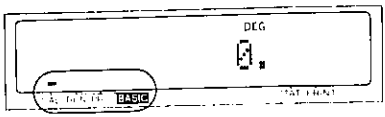
Beim **PC-1401/1402** werden drei Betriebsarten unterschieden:

1. CAL-Mode: Der Computer wird als wissenschaftlicher Rechner verwendet.

2. RUN-Mode: Gespeicherte Programme können gestartet werden. Zusätzlich kann der Computer im RUN-Mode auch als gewöhnlicher Taschenrechner ohne Programmunterstützung verwendet werden.

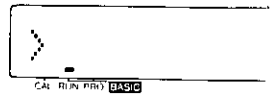
3. PRO-Mode: BASIC-Programme können eingegeben, gelistet und verändert werden.

Die Wahl der Betriebsart erfolgt mit den grünen Tasten [CAL] und [BASIC]
Die gewählte Betriebsart wird am Display durch einen kleinen waagrechten Strich (-) angezeigt.



Wird der **PC-1401/1402** eingeschaltet, so ist automatisch die Betriebsart CAL (culation) gewählt. Im Display wird das wie dargestellt angezeigt:

Wenn Sie nun die grüne Taste [BASIC] drücken, so wird die Betriebsart in den RUN-Mode umgeschaltet.



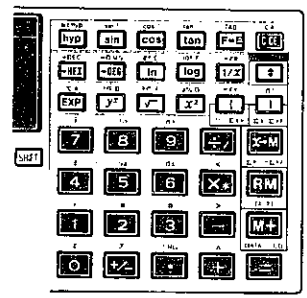
Durch nochmaliges Drücken der Taste [BASIC] wird in den PRO-Mode umgeschaltet.

Die [BASIC] -Taste ist ein Wechselschalter, d.h., durch einmaliges Drücken wird die Betriebsart umgeschaltet, durch nochmaliges Drücken der Urzustand wieder hergestellt.

Durch Drücken der [CAL] -Taste kann der Rechner wieder in den CAL-Mode umgeschaltet werden.

1.1.6.1 CAL-Mode

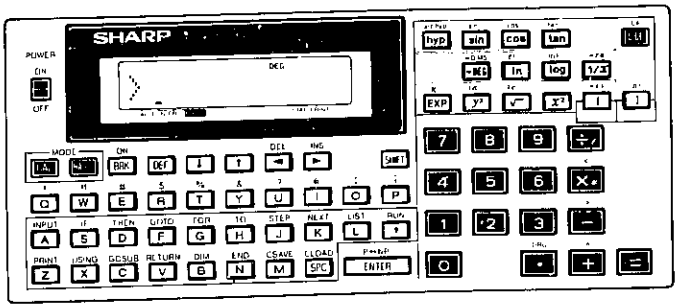
Schalten Sie Ihren Rechner durch Drücken der [CAL] -Taste in den CAL-Mode.
In diesem Mode können Sie mit den rechts abgebildeten Tasten den **PC-1401/1402** als wissenschaftlichen Rechner verwenden.



Beispiel:

[C-CE]	→	0.
[1] [2]	→	12.
[+]	→	12.
[3]	→	3.
[=]	→	15.

1.1.6.2 RUN- und PRO-Mode



Schalten Sie Ihren Rechner durch Drücken der Taste [BASIC] in den RUN- bzw. in den PRO-Mode. In dieser Betriebsart können die oben abgebildeten Tasten zur Eingabe verwendet werden.

Beispiele:

[PRINT Z] [USING X] [GOSUB C] → ZXC_

[r 1] [a 2] [DRG •] [b 3] → ZXC12.3_

[CA C·CE] → >

[INPUT A] [=] [x̄ 4] [^ +] [Sx 5] → A = 4 + 5_
↑
Cursor

[CA C·CE] [SHIFT] [PRINT Z] → PRINT_

[SHIFT] ["W] [1] [√] → PRINT "(√_

1.2 GRUNDLAGEN SHARP CE-126P

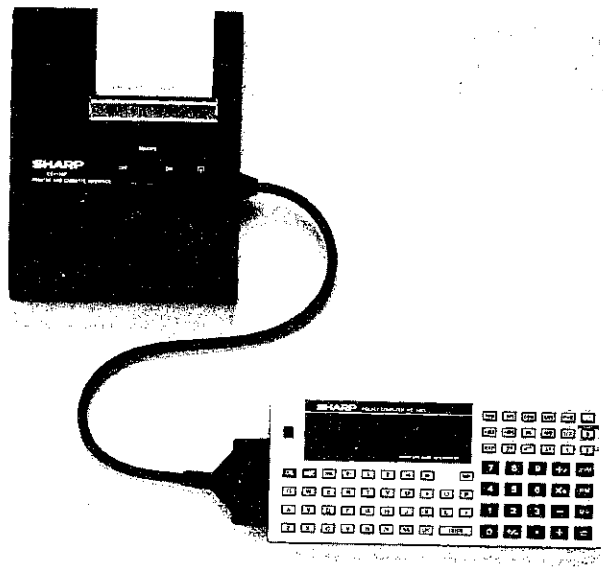
1.2.1 ALLGEMEINES

Die Option **SHARP CE-126P** (Thermodrucker mit integriertem Kassetten-Interface) ermöglicht es Ihnen, Ihre mit dem **PC-1401/1402** erstellten Programme oder Daten auszudrucken bzw. mit einem externen Kassettenrecorder zu speichern.

Eigenschaften des CE-126P:

- 24-Zeichen-Thermodrucker
- Manueller oder Programmgesteuerter Betrieb des Druckers
- 300-Baud-Kassetten-Interface
- Manueller oder Programmgesteuerter Betrieb des Kassetten-Interface
- Trockenbatterie-Betrieb (Transportfähigkeit)/Netzadapterbetrieb

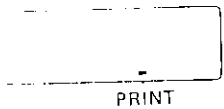
Der Anschluß des **CE-126P** an Ihren **PC-1401/1402** ist in dem, dem **CE-126P** beigelegten Handbuch beschrieben.



1.2.2 VERWENDUNG DES DRUCKERS

1.2.2.1 Manueller Druckbetrieb (Protokollerstellung)

Der manuelle Druckbetrieb ermöglicht es Ihnen, Ihre Rechnungen sofort auf Papier zu protokollieren. Dieser Betrieb wird durch Drücken der [SHIFT]- und der [ENTER]-Taste eingeschaltet. Am Display wird dies durch einen kleinen waagrechten Strich oberhalb von 'PRINT' angezeigt.



Auf dem Thermopapier wird bei der Protokollführung die gleiche Zeichenfolge in derselben Form, wie am Display angezeigt, ausgedruckt.

Nach Abschluß der Eingabe mit [ENTER] wird der Druck gestartet. Vorher können Sie Ihre Eingaben, wenn notwendig, noch korrigieren.

Das folgende Bild zeigt eine Protokollführung als Beispiel:

```
12000*.065
                          780.
780+25.6
                         805.6
SIN 45
          7.071067812E-01
F=50
                           50.
X=2*π*F
                  314.1592654
P=60*X
                  18849.55592
P/32
                  589.0486225
```

1.2.2.2 Programmgesteuerter Ausdruck

Enthalten BASIC-Programme LPRINT-Anweisungen, so können die den Anweisungen folgenden Werte oder Textausdrücke über den angeschlossenen Thermodrucker (Option **CE-126P**) ausgedruckt werden.

Programme, die mit PRINT-Anweisungen geschrieben wurden, können ganz einfach umgewandelt werden, so daß die Ausgabe nicht über das Display, sondern über den angeschlossenen Drucker erfolgt. Dies wird mit dem Befehl PRINT = LPRINT erreicht, der in einer Programmzeile dem Programm vorangestellt oder im RUN-Mode direkt eingegeben werden kann.

Im letzteren Fall muß das Programm jedoch mit [DEF] - key oder GOTO <Zeilennummer> gestartet werden.

Mit dem Befehl PRINT = PRINT kann dieser Befehl wieder aufgehoben werden.

Die Umschaltung kann auch in Abhängigkeit von einem logischen Vergleichsausdruck innerhalb einer IF-Anweisung erfolgen.

Das im Hauptspeicher gespeicherte Programm kann mittels der LLIST-Anweisung am Drucker gelistet werden. Näheres hierzu siehe LLIST-Anweisung, Seite 154.

Ist die zu druckende Programmzeile länger als 24 Zeichen, so erfolgt die Ausgabe am Drucker zwei- bzw. mehrzeilig. Dabei wird der Druck ab der zweiten Zeile um vier bzw. um sechs Stellen eingerückt, um den Ausdruck übersichtlicher zu gestalten.

Hinweise:

1. Wenn während des Drucks ein Fehler auftritt, der im Zusammenhang mit nicht richtig transportiertem Papier steht, so reißen Sie das Papier an der Abrißkante ab und ziehen das verbleibende Papier aus dem Drucker.
   Löschen Sie die Fehlermeldung durch Drücken der [C·CE] -Taste und legen Sie die Papierrolle wieder richtig ein.

2. Wenn der Drucker durch irgendwelche externe elektrische Störfelder beeinflußt wird, kann es sein, daß der Ausdruck fehlerhaft ist. Drücken Sie die [BRK] -Taste, um den Druck zu unterbrechen. Schalten Sie den Thermodrucker **CE-126P** aus und nach einigen Sekunden wieder ein. Initialisieren Sie den Drucker durch Drücken von [C·CE] .

3. Schalten Sie den Drucker nur ein, wenn Sie ihn benützen (Schonung der Batterien).

1.2.3 Verwendung des integrierten Kassetten-Interfaces

Das integrierte Kassetten-Interface ermöglicht es Ihnen, Ihre Programme bzw. Daten auf Band zu speichern. Dazu benötigen Sie einen für Datenaufzeichnung geeigneten Kassettenrekorder,z.B. den CE-152.

Einmal gespeicherte Programme oder Daten können ganz leicht wieder in den Rechner geladen werden.

1.2.3.1 Auswahl des Kassettenrekorders

Wir empfehlen Ihnen, für Ihre Programmspeicherung den extra dafür vorgesehenen Kassettenrekorder CE-152 zu verwenden. Der CE-152 wurde eigens zur Speicherung der mit dem SHARP-Taschencomputer PC-1401 erstellten Programme oder Daten entwickelt.

Wenn Sie jedoch einen anderen Rekorder verwenden wollen, sollte dieser folgende Ausstattung haben:

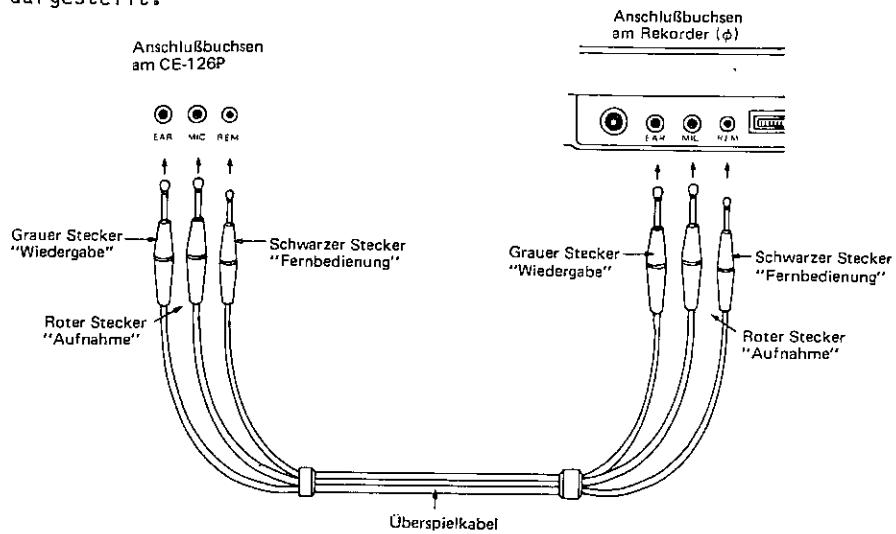
- Fernbedienung (REMOTE), Klinkenbuche 2,5 mm ø
- Bandzählwerk
- Mikrofoneingang (MIC), Klinkenbuchse 3,5 mm ø
- Ausgang (EAR), Klinkenbuchse 3,5 mm ø

Daneben sind folgende technische Daten gefordert:

- Eingangsimpedanz (MIC)	200...1000 Ohm
- Eingangsempfindlichkeit	Kleiner als 3 mV (-50 dB)
- Ausgangsimpedanz (EAR)	Kleiner als 10 Ohm
- Ausgangsleistung (EAR)	Größer als 1 V_{SS}
- Klirrfaktor	Kleiner als 15 % im Bereich zwischen 2...4 kHz
- Gleichlaufschwankungen	0,3 % max. (W.R.M.S)

1.2.3.2 Anschluß des Kassettenrekorders an das CE-126P

Die herzustellenden Verbindungen sind in der folgenden Abbildung dargestellt.



1.2.4 DATENSPEICHERUNG AUF MAGNETBAND

1.2.4.1 Allgemeines

Auf dem Magnetband können Sie Daten und Programme abspeichern oder von Band zurückladen. Außerdem können Sie das Magnetband als externen Programmspeicher verwenden um große Programme ablaufen zu lassen.

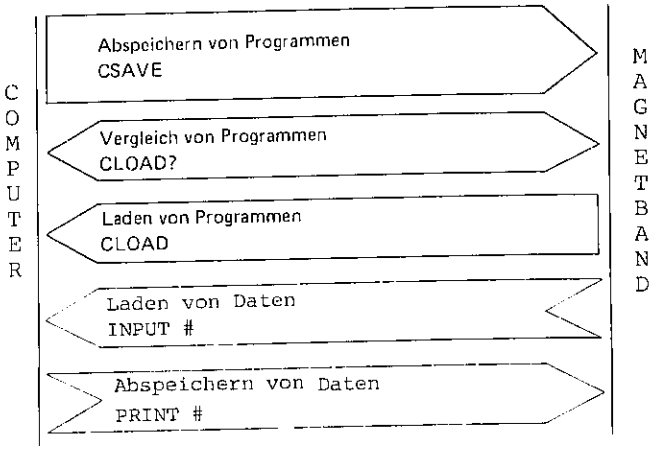
Die Informationen aus Daten und Programmen werden in Blöcken nach dem Zweitonverfahren auf das Magnetband geschrieben.

Auf einem Magnetband können abhängig vom Kassettentyp sehr viele Programme gespeichert werden. Damit Sie die Daten oder Programme wiederfinden und unterscheiden können, beginnt jeder Block aus einem Blocknamen (Programmnamen), der aus maximal 7 Zeichen besteht. Abgeschlossen wird der Speichervorgang, wenn die gesamte Information abgespeichert ist.

Um eine sichere Trennung der Blöcke zu gewährleisten schreibt der Rechner vor dem Blocknamen automatisch etwa 5 bis 7 Sekunden lang einen konstanten Signalton. Ein kompletter Block hat damit folgende Form:

	Signalton	Blockname	Programme oder Daten	

Folgende Möglichkeiten werden bei der Datenspeicherung auf Magnetband geboten:



Hinweis:

Auf die Informationen auf Band können Sie nur sequentiell zugreifen. Damit Sie Ihre Daten und Programme schneller wiederfinden, sollten Sie sich vor dem Abspeichern Informationen über den Zählerstand des Kassettenrekorders und den Blocknamen der Daten aufschreiben. Sie werden sehr schnell eine umfangreiche Programmbibliothek von Daten erhalten.

1.2.4.2 Speichern von Daten und Programmen

Schließen Sie Ihren Kassettenrecorder wie in Abschnitt 1.2.3.2 beschrieben an Ihr CE-126P an. Nachdem Sie Ihr Programm eingegeben haben, können Sie den Rekorder für die Abspeicherung vorbereiten:

- Stellen Sie den REMOTE-Schalter des Interfaces auf OFF.
- Legen Sie eine Kassette in den Rekorder ein.
- Vergewissern Sie sich, daß die eingelegte Kassette zurückgespult ist.
- Suchen Sie eine freie Stelle am Band.
  Merken Sie sich den Zählerstand.
- Stellen Sie den REMOTE-Schalter des Interfaces auf ON.
- Bereiten Sie die Aufnahme vor:
  "RECORD"- und "PLAY"-Taste drücken. Lautstärkeregler auf Mitte bis Maximum stellen, Tonhöhenregler auf "Höhen".
- Kommando Eingabe:
  Mit der CSAVE-Anweisung können Sie Ihr Programm gleichzeitig mit einem Namen, dem Blocknamen (Programmnamen) versehen.

Wenn Sie die [ENTER] -Taste betätigt haben, setzt sich das Magnetband in Bewegung. Ihr Programm wird jetzt übertragen und unter dem angegebenen Namen abgespeichert. Zu Beginn hören Sie 5 bis 7 Sekunden den konstanten Signalton, anschließend eine Folge von Tönen. Sobald der Computer am Ende des Programms angelangt ist, stopt das Band. Das Bereitschaftssymbol wird auf der Anzeige des **PC-1401/1402** angezeigt.

1.2.4.3 Überprüfen der Abspeicherung

Nach der Abspeicherung Ihres Programms mit der CSAVE-Anweisung und bevor Sie das Programm im Computer löschen, ist es empfehlenswert, zu überprüfen, ob es fehlerfrei übertragen wurde. Das Band könnte beispielsweise eine schadhafte Stelle haben.

Die Überprüfung ist ganz einfach und erfolgt mit der CLOAD?-Anweisung. Der Computer vergleicht das Programm in seinem Speicher mit dem auf dem Band gespeicherten Informationen. Ist die Aufzeichnung fehlerfrei, wird nach Abschluß der Überprüfung am Display des **PC-1401/1402** wieder das Bereitschaftszeichen angezeigt.
Stellt der Computer beim Vergleich der Informationen einen Fehler fest, erfolgt die Fehlermeldung ERROR 8.
Löschen Sie das aufgezeichnete Programm und versuchen Sie es nochmals zu speichern. Gegebenenfalls verwenden Sie eine andere Bandstelle bzw. eine andere Kassette.

1.2.4.4 Laden von Programmen oder Daten

Beim Laden von Programmen gehen Sie wie folgt vor:

- Lautstärkeregler auf Mitte bis Maximum stellen; Tonhöhenregler auf "Höhen".
- Spulen Sie das Band an den Anfang zurück.
- Stellen Sie das Zählwerk auf Null.
- Suchen Sie mit Hilfe des Zählwerks die Bandstelle, an der Ihr Programm beginnt.
- Schalten Sie den REMOTE-Schalter des Kassetten-Interface auf ON.
- Drücken Sie die "PLAY"-Taste Ihres Kassettenrekorders.
- Geben Sie ein:

```
CLOAD "PROG. 1"
```

Wenn Sie die [ENTER]-Taste gedrückt haben, setzt sich das Magnetband in Bewegung. Sie hören die gleiche Signalfolge wie bei der Abspeicherung. Am Display wird während des Ladevorgangs ein Asterix (*) angezeigt.

Bei der Eingabe der CLOAD-Anweisung gibt es zwei Möglichkeiten:

CLOAD oder CLOAD "Programmname"

Dabei wird das Programm vom Magnetband in den Speicher des Rechners übertragen. Die beiden CLOAD-Anweisungen unterscheiden sich nur dadurch, daß bei der zweiten Anweisung der Blockname (Programmname) zusätzlich mit eingegeben wird. Damit wird erreicht, daß nur jenes Programm, dessen Blockname mit dem eingegebenen übereinstimmt, übertragen wird. Alle anderen auf Band gespeicherten Programme werden überlesen.

T E I L II

P R A X I S

TEIL II P R A X I S

2.1 SHARP PC-1401/1402 - EINSATZ ALS WISSENSCHAFTLICHER RECHNER

2.1.1 BETRIEBSHINWEISE

Die folgenden Absatze enthalten einige wichtige Hinweise, die sich auf die CAL-Betriebsart des Rechners beziehen. Die in TEIL I, Abschnitt 1.1.2 enthaltenen allgemeinen Betriebshinweise haben darüber hinaus Gültigkeit.
Das Umschalten des Rechners in die CAL-Betriebsart ([CAL] -Taste drücken) sowie das für diese Betriebsart relevante Tastenfeld wurden bereits in TEIL I, Kapitel 1.2 aufgezeigt, so daß an dieser Stelle auf eine nahere Erläuterung dieser Punkte verzichtet werden kann.

2.1.1.1 Zweite Funktionsebene der Tastatur; Merkmale; Editier- und Korrekturmöglichkeiten

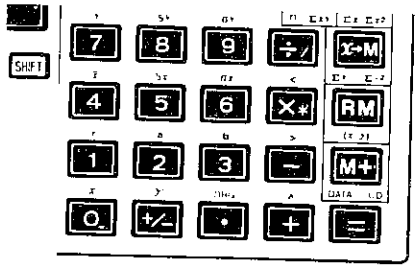
Wie aus der nachfolgenden Abbildung ersichtlich, weisen nahezu alle Tasten des Tastenfelds der CAL-Betriebsart eine zweite und, wenn man die Tastenfunktionen für die statistischen Berechnungen (STAT) und die Umwandlung "Hexadezimal-Dezimal" hinzuzählt, eine dritte und vierte Funktionsebene auf.

In den Erläuterungen und Rechenbeispielen ist die Auswahl der zweiten Tastenfunktion aus der Verwendung der [SHIFT]-Taste ersichtlich. Die Tastenfunktionen für die statistischen Funktionen und die Umwandlung "Hexadezimal-Dezimal" werden automatisch, d.h. durch das Drücken der [SHIFT] [STAT] -Taste bzw. [HEX] -Taste vorgewählt.

Bei näherer Betrachtung des Tastenfelds der CAL-Betriebsart wird bereits ein wichtiges Merkmal des **SHARP PC-1401/1402** deutlich. Es steht eine große Anzahl von Tastenfunktionen zur Verfügung, deren Verwendung zu einer wesentlichen Arbeitserleichterung und Arbeitsvereinfachung beiträgt.

Die Tastenfunktionen sind nachfolgend kurz aufgeführt und die entsprechenden Tasten angegeben. In den folgenden Abschnitten sind die einzelnen Funktionen zudem anhand von Berechnungsbeispielen erläutert. Die in Klammern stehende Seitenzahl gibt an, auf welcher Seite das Berechnungsbeispiel zu finden ist.

Grundrechenfunktionen (Seite 49):
Addition, Subtraktion, Multiplikation, Division (Vorzeichenumkehr)



Speicherrechnung (Seite 52):
Übertragen in den Speicher

x→M

Anzeigen des Speicherinhalts

RM

Addition und Subtraktion zum/vom Speicher

M+

Trigonometrische und Hyperbel-Funktionen (Seite 55):

hyp sin cos tan

Trigonometrische und inverse trigonometrische Funktionen (Arcusfunktionen)

	sin	cos	tan
SHIFT	sin⁻¹	cos⁻¹	tan⁻¹

Hyperbelfunktion und inverse Hyperbelfunktion (Areafunktion)

	hyp	sin	cos	tan
SHIFT	arc hyp	sin⁻¹	cos⁻¹	tan⁻¹

Umwandlung "Fließkomma-Wissenschaftliche" Schreibweise (Notation z.B. 1000 = 1×10^3) und Festlegung der Nachkommastellen bzw. der automatischen Aufrundung:

TAB
F↔E

Umwandlung "Fließkomma-Wissenschaftliche" Schreibweise (Seite 44)

F↔E

Festlegung Nachkommastellen/autom. Rundung (Seite 43)

SHIFT TAB (Eingabe Nachkommastellen)

Löschen der letzten Eingabe bzw. einer Fehlermeldung (Seite 42):

Löschen der vorangegangenen Eingabe bzw. des Displayinhalts durch Drücken der roten [C·CE]-Taste. Bei zweimaligem Drücken wird alles außer Speicher "M" gelöscht. Löschen von Speicher "M" durch Drücken von [C·CE] und [x→M].
Die Funktion "CA" ist in Betriebsart "CAL" unwirksam.

Umwandlung Dezimal-Hexadezimal und Hexadezimal-Dezimal (Seite 46):

Dezimal-Hexadezimal

[→HEX]

Hexadezimal-Dezimal

[→HEX] (Eingabe) [SHIFT] [→DEC]

Umwandlung von Winkel/Zeit (Seite 45):

Grad°/Minuten'/Sekunden'' in Dezimalzahl (Sexagesimal-Dezimal)
(Std./Min./Sek. in Dezimalzahl)

[→DEG]

Dezimalzahl in Grad°/Minuten'/Sekunden'' (Dezimal-Sexagesimal)
(Dezimalzahl in Std./Min./Sek.)

[SHIFT] [→D.MS]

Natürlicher Logarithmus und Exponentialfunktion der Zahl "e", Hexadezimaleingabe:

Natürlicher Logarithmus (Seite 56)

[ln]

Exponentialfunktion der Zahl "e" (Seite 57)

[SHIFT] [e^x]

Hexadezimal "E"

[→HEX] [E]

Allgemeiner Logarithmus und Exponentialfunktion 10^X, Hexadezimaleingabe:

[log]

Allgemeiner Logarithmus (Seite 56)

[log]

Exponentialfunktion 10^X (Seite 57)

[SHIFT] [10^x]

Hexadezimal "F"

[→HEX] [F]

Reziprok-Rechnen und Umwandlung von rechtwinkligen Koordinaten in Polarkoordinaten:

[1/x]

Reziprok-Rechnen (Seite 52)

[1/x]

Umwandlung von rechtwinkligen Koordinaten in Polarkoordinaten (Seite 46)

[SHIFT] [→r,θ]

Austausch der im Rechner gespeicherten Zahl (Ersteingabe, Ergebnis) mit der angezeigten Zahl und Anwahl der Betriebsart "Statistische Berechnungen":

[↕]

Austausch (Seite 46)

[↕]

Anwahl der Betriebsart "Statistische Berechnungen" (Seite 57)

[SHIFT] [STAT]

Exponenteingabe (wissenschaftliche Schreibweise) und Eingabe der Kreiszahl (Pi), Hexadezimaleingabe:

[EXP]

Exponenteingabe (Seite 44)

[EXP]

Kreiszahl (Pi)

[SHIFT] [π]

Hexadezimal "A"

[→HEX] [A]

Potenzieren und Berechnung der x-ten Wurzel von y, Hexadezimaleingabe (Seite 51):

[y^x]

Potenzieren

[y^x]

Wurzelziehen

[SHIFT] [$\sqrt[x]{y}$]

Hexadezimal "B"

[HEX] [B]

Berechnung von Quadrat- und Kubikwurzel, Hexadezimaleingabe (Seite 51):

[$\sqrt{}$]

Quadratwurzel

[$\sqrt{}$]

Kubikwurzel

[SHIFT] [$\sqrt[3]{}$]

Hexadezimal "C"

[HEX] [C]

Quadrierung und Prozentberechnung, Hexadezimaleingabe (Seite 51):

[x^2]

Quadrierung der angezeigten Zahl

[x^2]

Prozentberechnung (Prozentsatz, prozentuale Zunahme)

[SHIFT] [%]

Hexadezimal "D"

[HEX] [D]

Öffnen der Klammer und Umwandlung von Polarkoordinaten in rechtwinklige Koordinaten, statistische Berechnungen:

[(]

Öffnen der Klammer (Seite 53)

[(]

Umwandlung von Polarkoordinaten in rechtwinklige Koordinaten (Seite 46)

[SHIFT] [$\rightarrow xy$]

Statistische Berechnungen (Seite 57)
Datenanzahl
"STAT" [n]
Datensumme von xy
"STAT" [SHIFT] [Σxy]

Schließen der Klammer und Fakultätswert, statistische Berechnungen:

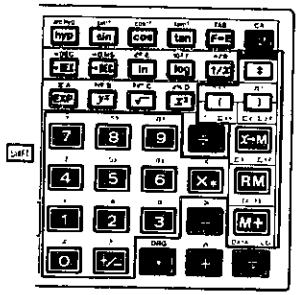
[ )]

Schließen der Klammer (Seite 53)
[)]

Fakultätswert (Seite 57)
[SHIFT] [n!]

Statistische Berechnungen (Seite 57)
Datensumme von x
"STAT" [Σx]
Summe von x^2
"STAT" [SHIFT] [Σx²]

Statistische Berechnungen (Seite 57):



Anwahl der Betriebsart "Statistische Berechnungen"
[SHIFT] [STAT]
Operandenanzahl
[n]
Datensumme von xy
[SHIFT] [Σxy]
Datensumme von x
[Σx]
Summe von x^2
[SHIFT] [Σx²]

Mittelwert der Daten y
[SHIFT] [ȳ]

Standardabweichung der y-Werte mit Gesamtheitsparameter "n-1"
[SHIFT] [sy]

Standardabweichung der y-Werte mit Gesamtheitsparameter "n"
[SHIFT] [σy]

Datensumme von y
[Σy]

Summe von y^2
[SHIFT] [Σy²]

Mittelwert der Daten x
[SHIFT] [x̄]

Standartabweichung der x-Werte mit Gesamtheitsparameter "n-1"
[SHIFT] [Sx]

Standartabweichung der x-Werte mit Gesamtheitsparameter "n"
[SHIFT] [σx]

Eingabe von x- und y-Daten
[x,y]

Korrelationskoeffizient
[SHIFT] [r]

Schnittpunkt der Regressionslinie mit der y-Achse
[SHIFT] [a]

Steilheit der Regressionslinie
[SHIFT] [b]

Dateneingabe
[DATA]

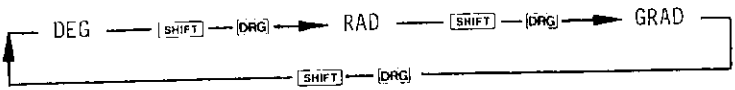
Korrektur einer falschen Eingabe
[SHIFT] [CD]

Näherungswert von x bei vorgegebenem y-Wert
Eingabe y-Wert [SHIFT] [x']

Näherungswert von y bei vorgegebenem x-Wert
Eingabe x-Wert [SHIFT] [y']

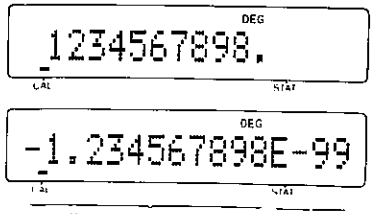
Anwahl der Winkeleinheit (Seite 40):
DEG (Grad; °, ', '')
RAD (Radiant; rad)
[SHIFT] [DRG]

GRAD (Neugrad; gon)
[SHIFT] [DRG]
[SHIFT] [DRG]



Anmerkung:
Die Funktionen (CA), [<], [>] und [^] werden in der Betriebsart "CAL" nicht verwendet.

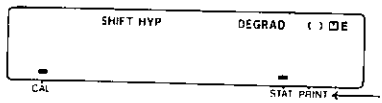
2.1.1.2 Anzeige/Display-Format und Symbole.



Fließkommaschreibweise (Normalformat)

Wissenschaftliche Schreibweise

Die Anzeige des **PC-1401/1402** umfaßt 16 Stellen. In der Betriebsart "CAL" werden die Rechenergebnisse soweit möglich in Fließkommaschreibweise dargestellt. Ist das Rechenergebnis kleiner 0,000000001 oder größer 9999999999 (größer -0,000000001 oder kleiner -9999999999), erfolgt die Anzeige in wissenschaftlicher Schreibweise. Dabei wird die Mantisse 12stellig (einschließlich Vorzeichen und Komma) und der Exponent 4stellig (einschließlich Symbol und Vorzeichen) angezeigt.



Ausdruck über Drucker siehe 1.2 "GRUNDLAGEN SHARP CE-126P"

Die Betriebsart "CAL" verwendet die oben abgebildeten Symbole und Markierungen, die nachfolgend aufgelistet sind und deren Bedeutung erläutert wird.

CAL Die Markierung (-) über der Beschriftung "CAL" gibt an, daß sich der Rechner in der Betriebsart "CAL" befindet.

SHIFT Das Wort "SHIFT" wird angezeigt, wenn die [SHIFT] -Taste gedrückt wurde und damit die zweite Funktionsebene der Tasten angewählt ist.

HYP Das Wort "HYP" wird angezeigt, wenn die [hyp] -Taste gedrückt wurde und damit eine Hyperbelfunktion angewählt ist. Werden die Tasten [SHIFT] und [hyp] gedrückt (Anzeige "SHIFT HYP"), so ist damit eine inverse Hyperbelfunktion (Areafunktion) angewählt.

DEG
RAD
GRAD Diese drei Wörter können durch wiederholtes Betätigen der Tasten [SHIFT] und [DRG] angewählt werden. Die Wörter geben an, in welcher Winkeleinheit gerechnet wird.

DEG : (Grad; °, ', '')
RAD : (Radiant; rad)
GRAD: (Neugrad; gon)
Gestreckter Winkel = 180° = π rad = 200 gon

( ) Das Klammersymbol erscheint, sobald beim Eintippen einer Berechnungsformel die [(] -Taste verwendet wird.

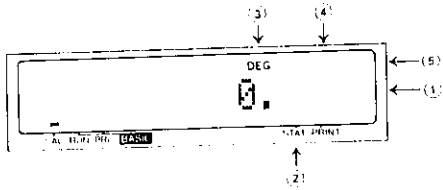
M Dieses Symbol zeigt an, daß sich ein numerischer Wert, der ungleich Null ist, im Speicher befindet und der Speicher somit belegt ist.

E Das Symbol "E" in der rechten oberen Ecke des Anzeigefelds zeigt an, daß eine Bereichsüberschreitung aufgetreten ist oder eine falsche Anweisung gegeben wurde. Der Fehlerzustand kann durch das Betätigen der [C·CE] -Taste beseitigt werden.

STAT Die Markierung (-) über der Beschriftung "STAT" im rechten unteren Bereich des Anzeigefelds gibt an, daß sich der Rechner in der Betriebsart "Statistische Berechnungen" befindet. Die Betriebsart wird durch Drücken der Tasten [SHIFT] und [STAT] angewählt und durch nochmaliges Drücken wieder ausgeschaltet.

2.1.1.3 Grundeinstellung

Bevor mit den Berechnungen begonnen wird, sollte die Grundeinstellung des Rechners überprüft werden. Dazu den POWER-Schalter auf ON schalten oder, falls der Rechner automatisch abgeschaltet hat, die [ON BRK] -Taste drücken. Anschließend die [CAL] -Taste (grün) drücken, um so die Betriebsart "CAL" anzuwählen und zweimal die [C·CE] -Taste betätigen. Das Anzeigefeld sollte nun, entsprechend der nachfolgenden Abbildung, nur die Anfangsinformationen enthalten.



Falls dies nicht der Fall ist, die folgenden Kurzanweisungen lesen und die erforderlichen Maßnahmen durchführen.

(1) Es wird mehr als eine Null angezeigt (z.B. 0.000):
Die Anzahl der Nachkommastellen ist festgelegt. Durch Drücken der Tasten [SHIFT] [TAB] [.] auf Fließkomma mit vollem Anzeigeformat umschalten.

② Über der Beschriftung "STAT" wird eine Markierung (-) angezeigt:
Die Betriebsart "Statistische Berechnungen" ist angewählt.
Durch Drücken der Tasten [SHIFT] [STAT] die Betriebsart aufheben.

③ Statt DEG wird RAD oder GRAD angezeigt:
Die Winkeleinheit ist in Radiant (rad) oder Neugrad (gon) vorgewählt. Durch Drücken der Tasten [SHIFT] [DRG] Winkeleinheit auf DEG einstellen.

④ Das Symbol M wird angezeigt:
Der Speicher ist mit einem numerischen Wert belegt. Löschen des Speichers durch Drücken der Tasten [C·CE] [x→M] .

⑤ Alle im oberen Bereich des Anzeigefelds dargestellten Symbole, mit Ausnahme der in Punkt ③ und ④ beschriebenen, können durch Drücken der [C·CE] -Taste gelöscht werden.

2.1.1.4 Löschen der Eingabe oder einer Fehlermeldung

Durch einmaliges Drücken der roten [C·CE] -Taste wird die letzte Eingabe bzw. der Displayinhalt gelöscht. Ein zweimaliges Drücken der [C·CE] -Taste löscht auch den zuerst eingegebenen Wert bzw. das Ergebnis der vorangegangenen Berechnung.
Der Speicherinhalt M wird dadurch nicht gelöscht.

123 [+] 456	456.
[C·CE]	0.
789 [=]	912.

(123 + 789 = 912)

6 [×] 2 [+]	12.
[C·CE]	0.
6 [÷] 2 [+]	3.
5 [=]	8.

Die [C·CE] -Taste dient darüber hinaus zum Löschen der Fehlermeldung (E).

5 [÷] 0 [=]	0. E	← Fehler (Error)-Symbol
[C·CE]	0.	

2.1.1.5 Auf-/Abrundung bzw. Festlegung der Nachkommastellen

Die Auf- oder Abrundung des Rechenergebnisses erfolgt in der Regel in der 11. Stelle. Da das Display 10 Stellen aufweist, ist die letzte Stelle des angezeigten Rechenergebnisses nicht auf- oder abgerundet. Bei Benützung der Tasten [=] , [M+] oder [M-] wird der angezeigte, nicht gerundete Wert im Falle weiterer Berechnungen verwendet.
Beispiel:

1	[÷]	3	[×]	3	[=] 1.
		1	[÷]	3	[=] 0.333333333
			[×]	3	[=] 0.999999999

Die Anzahl der Nachkommastellen und damit die Rundung kann frei festgelegt werden. Dazu dient die [TAB]-Taste in Verbindung mit den Tasten (0) bis (9) und (.). Die Festlegung der Nachkommastellen wird durch Drücken der Tasten [SHIFT] [TAB] und [.] aufgehoben.

[SHIFT]	[TAB]	[0]	→	Keine Nachkommastellen. Entsprechend der 1. Nachkommastelle wird auf- oder abgerundet.
[SHIFT]	[TAB]	[1]	→	Eine Nachkommastelle. Entsprechend der 2. Nachkommastelle wird auf- oder abgerundet.
	⋮			
[SHIFT]	[TAB]	[9]	→	Neun Nachkommastellen. Entsprechend der 10. Nachkommastelle wird auf-oder abgerundet.
[SHIFT]	[TAB]	[.]	→	Rückstellung auf Fließkomma. Entsprechend der 12. Stelle bzw. der 10. Nachkommastelle bei wissenschaftlicher Schreibweise wird auf- oder abgerundet.

Beispiel: [SHIFT] [TAB] [9] [.] [5] [÷] [9] [=] → 0.055555556
Neun Nachkommastellen. Die 9. Nachkommastelle wurde aufgerundet.

[F↔E] → 5.555555556E-02
Neun Nachkommastellen. Die 9. Nachkommastelle der Mantisse wurde aufgerundet.

[SHIFT] [TAB] [3] → 5.556E-02
Drei Nachkommastellen. Die 3. Nachkommastelle der Mantisse wurde aufgerundet.

F↔E → 0.056
Drei Nachkommastellen. Die 3. Nachkommastelle wurde aufgerundet.

SHIFT TAB • → 0.055555555
Der Wert ist im Rechner in der Form von $5.55555555555 \times 10^{-2}$ gespeichert.
Die Rundung in der 9. Nachkommastelle der Mantisse ergibt $5.555555556 \times 10^{-2}$.
Da das Display 10 Stellen anzeigt, ist die Aufrundung in der Ergebnisanzeige nicht enthalten.

F↔E → 5.555555556E-02

2.1.1.6 Wissenschaftliche Schreibweise (Notation)

Die Displayanzeige erfolgt, solange die Anzeigekapazität nicht unter- oder überschritten wird, in Festkommaschreibweise. Durch Drücken der F↔E -Taste kann jedoch auf einfache Weise zwischen "Festkomma-" und "Wissenschaftlicher Schreibweise" (Notation) gewählt werden.
Soll oder muß ein Wert in wissenschaftlicher Schreibweise eingegeben werden, ist zur Eingabe des Exponenten die EXP -Taste zu verwenden. Das Vorzeichen des Exponenten läßt sich mit der +/- -Taste bestimmen.

Beispiel:

23 × 1000 =	23000.
F↔E	2.3E 04
F↔E	23000.
4 π A EXP 3	4.E 03 (4×10^3)
=	4000.
1.23 +/-	−1.23
EXP 5 +/-	−1.23E−05 (-1.23×10^{-5})
=	−0.0000123

2.1.1.7 Umwandlung des Winkels und der Zeit

Zur Umwandlung von Winkel- oder Zeitangaben (°, ′, ″ bzw. Std., Min., Sek.) in den entsprechenden Dezimalwert muß der Wert als ganze Zahl (°; Std.) plus Nachkommastellen (′, ″; Min., Sek.; jeweils zweistellig) eingegeben werden.

Aufgabe: Umwandlung von 12° 47′ 52″ in den entsprechenden Dezimalwert
Tastenfolge: 12.4752 [DEG]
Ergebnis: 12.79777778

Bei der Umwandlung des dezimalen Winkel-/Zeitwerts in den sexagesimalen Wert (°, ′, ″ bzw. Std., Min., Sek.) ist das angezeigte Ergebnis wie folgt zuzuordnen:

Ganze Zahl	Nachkommastellen		
	1 und 2	3 und 4	5 bis 9
° (Std.)	′ (Min.)	″ (Sek.)	Zehntelsekunden

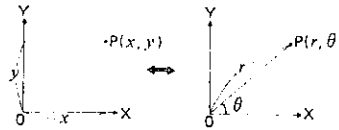
Aufgabe: Umwandlung des dezimalen Winkelwerts 24,7256 in den entsprechenden Sexagesimalwert
Tastenfolge: 24.7256 [SHIFT] [°'"]
Ergebnis: 24.433216 24°43′32″

Beispiel:
Ein Pferd läuft Rundenzeiten von 2 Min. 25 Sek., 2 Min. 38 Sek. und 2 Min. 22 Sek. Wie hoch ist die durchschnittliche Rundenzeit?

Tastenfolge: .0225 [DEG] [+] .0238 [DEG] [+] .0222 [DEG] [=]
Zwischenergebnis: 0.123611111
Tastenfolge: [1/x] 3 [=]
Zwischenergebnis: 0.041203703
Tastenfolge: [SHIFT] [°'"]
Ergebnis: 0.022833333
Die durchschnittliche Rundenzeit ist
2 Minuten 28 Sekunden

2.1.1.8 Koordinatenumwandlung

Umwandlung von rechtwinkligen Koordinaten in Polarkoordinaten $(x, y \rightarrow r, \theta)$



$r = \sqrt{x^2 + y^2}$

$\theta = \tan^{-1} \frac{y}{x}$

DEG: $0 \leqq |\theta| \leqq 180$
RAD: $0 \leqq |\theta| \leqq \pi$
GRAD: $0 \leqq |\theta| \leqq 200$

Aufgabe: Umwandlung der rechtwinkligen Koordinaten x = 6 und y = 4 in Polarkoordinaten. Winkeleinheit DEG (°).
Tastenfolge: 6 [↕] 4 [SHIFT] [→rθ]
Ergebnis: 7.211102551 (r)
Tastenfolge: [↕]
Ergebnis: 33.69006753 (θ)

Aufgabe: Umwandlung des Real- und Imaginärteils eines Vektors in Betrag und Richtung (Phase). Winkeleinheit DEG (°).
Realteil x = 12
Imaginärteil y = j9
Tastenfolge: 12 [↕] 9 [SHIFT] [→rθ]
Ergebnis: Betrag r = 15
Tastenfolge: [↕]
Ergebnis: Richtung (Phase) θ = 36.86989765

Umwandlung von Polarkoordinaten in rechtwinklige Koordinaten (Formel)
Aufgabe: Umwandlung der Polarkoordinaten r = 14 und $\theta = \pi/3$ in rechtwinklige Koordinaten. Winkeleinheit RAD (rad).
Tastenfolge: [SHIFT] [DRG]
[SHIFT] [π] [÷] 3 [=] [↕] 14 [↕] [SHIFT] [→xy]
Ergebnis: 7.000000002 (x)
Tastenfolge: [↕]
Ergebnis: 12.12435565 (y)

Anmerkung: Es wird zuerst $\pi/3$ errechnet. Nach der Eingabe von r = 14 werden die beiden Werte durch Drücken der [↕] -Taste ausgetauscht und die richtige Reihenfolge so wieder hergestellt.

2.1.1.9 Umwandlung "Hexadezimal - Dezimal" und Rechnen mit Zahlen in Hexadezimal-Schreibweise

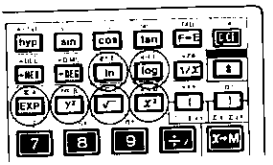
Die einfache Umwandlung von Zahlen zur Basis 10 (Dezimal) in Zahlen zur Basis 16 (Hexadezimal) und umgekehrt ist auf dem Feld der Datenverarbeitung und Computertechnik von großer Bedeutung, da in diesen Bereichen die Hexadezimal-Schreibweise häufig anzutreffen ist.

Die Hexadezimal-Schreibweise verwendet die Zahlen 0 bis 9 sowie die Buchstaben A bis F, wobei diese in Dezimal-Schreibweise den Werten 0 bis 15 entsprechen.

Dezimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Hexadezimal	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Zur Eingabe einer Zahl in Hexadezimal-Schreibweise muß zuerst die [HEX]-Taste gedrückt werden. Die Anzeige "HEX" weist darauf hin, daß es sich bei dem angezeigten Wert um eine Sedezimalzahl (Zahl in Hexadezimal-Schreibweise) handelt.

Die Buchstaben A bis F können nach Drücken der [HEX]-Taste über die in der nebenstehenden Abbildung hervorgehobenen Tasten eingegeben werden.



Das Umschalten auf Dezimal-Schreibweise erfolgt durch Drücken der Tasten [SHIFT] [DEC].

Umwandlung "Dezimal-Hexadezimal"

Aufgabe: Umwandlung von Dezimal 30 in Hexadezimal

Tastenfolge: 30 [HEX] Anzeige: 1E. HEX

Anmerkung: Zur Eingabe einer weiteren Dezimalzahl durch Drücken der Tasten [SHIFT] [DEC] wieder auf Dezimal-Schreibweise umschalten.

Aufgabe: Umwandlung von Dezimal -2 in Hexadezimal

Tastenfolge: [2] [+/-] [HEX] Anzeige: FFFFFFFFFE. HEX

Anmerkung: Die Umwandlung einer negativen Dezimalzahl erfolgt intern über das "Zweierkomplement". Das Resultat wird im "16er-Komplement" angezeigt.
Mit der [+/-] -Taste kann das Vorzeichen der angezeigten Zahl gewechselt werden. Bei Hexadezimal-Schreibweise erfolgt die Anzeige des jeweiligen 16er-Komplements.

Aufgabe: Umwandlung von Dezimal 123,4 in Hexadezimal

Tastenfolge: [SHIFT] [DEC] 123.4 [HEX] Anzeige: 7B. HEX

Anmerkung: Es können nur ganze Zahlen in Hexadezimal umgewandelt werden. Die Nachkommastellen bleiben bei der Umwandlung unberücksichtigt.

Umwandlung "Hexadezimal-Dezimal"

Aufgabe: Umwandlung von Hexadezimal 2BC in Dezimal

Tastenfolge: [C·CE] [•HEX] 2 B C [SHIFT] [•DEC]

Anzeige: 700.

Aufgabe: Umwandlung von Hexadezimal FFFFFFFF12 in Dezimal

Tastenfolge: [C·CE] [•HEX] FFFFFFFF 12 [SHIFT] [•DEC]

Anzeige: FFFFFFFF12. HEX

−238.

Anmerkung: Die Umwandlung von Hexadezimal FDABF41C01 bis FFFFFFFFFF ergibt eine negative Dezimalzahl.

Rechnen mit Zahlen in Hexadezimal-Schreibweise

In der Betriebsart "HEX" können auch Rechenoperationen durchgeführt werden. Dabei sind folgende Hinweise zu beachten:

- Es kann nur mit ganzen Zahlen gerechnet werden, die [.] -Taste ist daher ohne Funktion.
- Bei Divisionen wird immer eine ganze Zahl als Ergebnis angezeigt. Etwaige Nachkommastellen bleiben unberücksichtigt.
  Beispiel: B [÷] 3 [=] ... 3. HEX
- Falls mit einem Zwischenergebnis weitergerechnet wird, das Nachkommastellen aufweist, erfolgt eine Fehlermeldung (Symbol "E" wird angezeigt).
  Beispiel: B [÷] 3 [×] ... Error (Symbol "E")
- Bei Kettenrechnungen sind ggf. Klammern zu verwenden, da keine Beachtung der Vorrangordnung erfolgt.
- Durch Drücken der [+/−] -Taste kann auf einfache Weise das Komplement der angezeigten Sedezimalzahl ermittelt werden.
  Beispiel: A 8 [+/−] → FFFFFFFF55. HEX
  [+/−] → AB. HEX

Aufgabe: A4 + BA
Tastenfolge: A4 [+] B A [=]

Anzeige: 15 E. HEX

Aufgabe: 8 x 3
Tastenfolge: 8 [×] 3 [=]

Anzeige: 18. HEX

Aufgabe: (12 + D) x B
Tastenfolge: [C·CE] [(] 12 [+] D [)] [×] B [=]

Anzeige: 155. HEX

Aufgabe: 43A − 3CB =
+) A38 − 2FB =
(Total)

Tastenfolge: [C·CE] [x→M]
4 3 A [−] 3 C B [M+]

Anzeige: 6F. HEX

Tastenfolge: A 3 8 [−] 2 F B [M+]

Anzeige: 73D. HEX

Tastenfolge: [RM]

Anzeige: 7AC. HEX

2.1.2 NORMALE BERECHNUNGEN

Die nachfolgenden Beispiele zeigen die Bedienung des Rechners bzw. die Tastenfolge für die Grundrechenfunktionen.
Bevoe damit begonnen wird, die Beispiele nachzuvollziehen, sollte, um eine gemeinsame Ausgangsbasis sicherzustellen, eine kurze Überprüfung der Grundeinstellung des **PC-1401/1402** erfolgen.

POWER-Schalter auf ON schalten
(Falls der Rechner automatisch abgeschaltet hat, zum Einschalten die [ON BRK] -Taste drücken)
[CAL] -Taste (grün) drücken
[C·CE] -Taste (rot) zweimal drücken

Das Anzeigefeld sollte nun nur die Anfangsinformationen enthalten.
Ist dies nicht der Fall, die Grundeinstellung des PC-1401/1402 gemäß 2.1.1.3 (Seite 41) durchführen.

2.1.2.1 Addition und Subtraktion

Aufgabe: Rechnen von 123 + 654

Tastenfolge	Anzeige
r 1 a 2 b 3	123.
^ +	123.
σx 6 Sx 5 x̄ 4	654.
=	777.

(123 + 654 = 777)

Aufgabe: 12 + 45,6 - 32,1 + 789 - 741 + 213
Tastenfolge: 12 + 45.6 − 32.1 + 789 − 741 + 213 =
Ergebnis: 286.5

2.1.2.2 Multiplikation und Division

Aufgabe: 841 x 586 / 0,12
Tastenfolge: 841 × 586 ÷ .12 =
Ergebnis: 4106883.333

Aufgabe: 427 + 54 x 32 / 7 - 39 x 2
Tastenfolge: 427 + 54 × 32 ÷ 7 − 39 × 2 =
Ergebnis: 595.8571429

Anmerkung: Multiplikation und Division haben Vorrang vor Addition und Subtraktion. Die Vorrangordnung wird beachtet. Der **PC-1401/1402** führte intern zuerst die Multiplikation und Division durch (siehe auch 2.1.2.7, Seite 53).

Multiplikationen mit einer Konstanten
Der zuerst eingegebene Wert kann als Konstante verwendet werden.
Aufgabe: 5 x 3 = ? , 10 x 3 = ? , usw.
Tastenfolge: 3 × 5 =
Ergebnis: 15
Tastenfolge: 10 =
Ergebnis: 30

Divisionen mit einer Konstanten
Der als zweites eingegebene Wert kann als Konstante verwendet werden.
Aufgabe: 15 / 3 = ? , 30 / 3 = ? , usw.
Tastenfolge: 15 [÷] 3 [=]
Ergebnis: 5
Tastenfolge: 30 [=]
Ergebnis: 10

Hinweis: Bei Kettenrechnungen bleiben die entsprechend der Vorrangordnung letzte Recheninstruktion und der jeweilige numerische Wert erhalten und können damit für weitere Berechnungen bzw. als Konstanten verwendet werden.

Ausgeführter Rechenvorgang	Konstante	
$a+b\times c=$	$+bc$	(Konstanten-Addition)
$a\times b\div c=$	$\div c$	(Konstanten-Division)
$a\div b\times c=$	$\frac{a}{b}\times$	(Konstanten-Multiplikation)
$a\times b-c=$	$-c$	(Konstanten-Subtraktion)

2.1.2.3 Potenz- und Wurzelfunktion

Aufgabe: 20^2
Tastenfolge: 20 [x²]
Ergebnis: 400

Aufgabe: 3^3
Tastenfolge: 3 [y^x] 3 [=]
Ergebnis: 27

Aufgabe: 3^4
Tastenfolge: 3 [y^x] 4 [=]
Ergebnis: 81

Aufgabe: $\sqrt{25}$
Tastenfolge: 25 [√]
Ergebnis: 5

Aufgabe: 3. Wurzel von 27
Tastenfolge: 27 [SHIFT] [∛]
Ergebnis: 3

Aufgabe: 4. Wurzel von 81
Tastenfolge: 81 [SHIFT] [x√y] 4 [=]
Ergebnis: 3

2.1.2.4 Prozentrechnung

Aufgabe: 45% von 2780 $\left(2{,}780\times\frac{45}{100}\right)$
Tastenfolge: 2780 [×] 45 [SHIFT] [%]
Ergebnis: 1251

Prozentuale Zunahme/Abnahme
Der Ausgangswert (= 100%) muß als zweiter Wert, d.h. nach Drücken der [−] -Taste eingegeben werden.

Aufgabe: Anstieg von 473 auf 547, Prozentuale Zunahme ?
Tastenfolge: 547 [−] 473 [SHIFT] [Δ%]
Ergebnis: 15.6448203

Aufgabe: Rückgang von 547 auf 473, Prozentuale Abnahme ?
Tastenfolge: 473 [−] 547 [SHIFT] [Δ%]
Ergebnis: -13.52833638

2.1.2.5 Reziprok-Rechnung

Aufgabe: 1/6 + 1/7
Tastenfolge: 6 [1/x] [+] 7 [1/x] [=]
Ergebnis: 0.309523809

2.1.2.6 Speicherrechnung

Der zur Verfügung stehende Speicher wird über die blau beschrifteten Tasten [x→M] [RM] [M+] angesprochen. Bevor mit dem Speicher gearbeitet wird, muß dessen Inhalt ggf. erst gelöscht werden. Der Speicherinhalt bleibt auch bei ausgeschaltetem Rechner erhalten. Die Belegung des Speichers zeigt das Symbol M in der rechten oberen Ecke des Displays an. Zum Löschen des Speichers die Tasten [C·CE] und [x→M] drücken.

Das Addieren eines Rechenergebnisses oder eines eingegebenen Werts zum Speicherinhalt erfolgt durch Drücken der [M+] -Taste. Das Drücken der [RM] -Taste bringt den Speicherinhalt am Display zur Anzeige.

Tastenfolge	Anzeige
12 [+] 5 [M+]	17
[C·CE]	0
[RM]	17
8 [÷] 2 [M+]	4
[RM]	21 (Speicherinhalt)

Soll ein Rechenergebnis oder ein eingegebener Wert vom Speicherinhalt subtrahiert werden, muß vor dem Betätigen der [M+] -Taste durch Drücken der [+/−] -Taste das Vorzeichen gewechselt werden.

Tastenfolge	Anzeige
2 [+] 5 [=] [+/−] [M+]	-7
[C·CE]	0
[RM]	14

Das Drücken der [x→M] -Taste übertragt den im Display angezeigten Wert in den Speicher. Der Speicherinhalt wird überschrieben.

Tastenfolge	Anzeige
12 [×] 2 [=] [x→M]	24
[C-CE]	0
[RM]	24

2.1.2.7 Vorrangordnung und Verwendung der "Klammern"-Tasten

Die Verwendung der Tasten [(] und [)] ist dann zwingend notwendig, wenn eine Serie von Einzelberechnungen zu einem Rechenvorgang zusammengefaßt werden soll und dabei die Vorrangordnung der Algebra außer Kraft gesetzt werden muß.
Sobald die [(] -Taste gedrückt wird, erscheint in der rechten oberen Ecke des Anzeigefelds das ( )-Symbol, welches so lange sichtbar bleibt, bis alle geöffneten Klammern wieder geschlossen wurden. Das Schließen der Klammer(n) über die [)] -Taste kann unterbleiben, wenn unmittelbar folgend die Tasten [=] oder [M+] zu drücken sind, da diese die gleiche Funktion erfüllen.
Die Berechnungen in der Klammer haben Priorität vor allen anderen Berechnungen. Die Klammerfunktion kann in einer Ebene bis zu fünfzehnmal verwendet werden. Als erstes werden die Berechnungen der innersten Klammer ausgeführt.

Aufgabe: $12 + 42 \div (8 - 6)$
Tastenfolge: 12 [+] 42 [÷] [(] 8 [−] 6 [)] [=]
Ergebnis: 33

Aufgabe: $126 \div [(3 + 4) \times (3 - 1)]$
Tastenfolge: 126 [÷] [(] [(] 3 [+] 4 [)] [×] [(] 3 [−] 1 [)] [)] [=]
Ergebnis: 9
Anmerkung: Das Schließen der beiden Klammern unmittelbar vor der [=] -Taste (oder [M+] -Taste) kann unterbleiben.

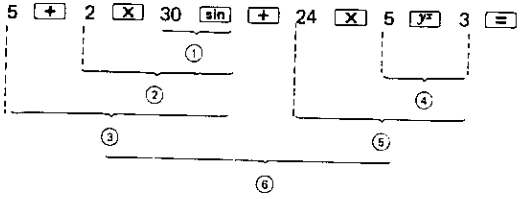
Vorrangordnung
Da der **PC-1401/1402** die Vorrangordnung der einzelnen Berechnungen beachtet, ist es möglich, mathematische Formeln direkt, d.h., ohne daß Klammern eingefügt oder Formelumstellungen vorgenommen werden müssen, einzugeben. Die Vorrangordnung bzw. die Reihenfolge, nach der die Berechnungen in einer langeren Rechenformel ausgeführt werden, ist wie folgt:

(1) Funktionen wie sin oder x^2
(2) y^x, $\sqrt[x]{y}$
(3) ×, ÷
(4) +, −
(5) =, M+, Δ%

(Berechnungen, die gleiche Priorität besitzen, werden der Reihe nach ausgeführt)

Beispiel: Tastenfolge und Reihenfolge der Berechnungen bei
$5 + 2 \times \sin 30 + 24 \times 5^3 = 3006$
Winkeleinheit DEG

Tastenfolge: 5 [+] 2 [×] 30 [sin] [+] 24 [×] 5 [y^x] 3 [=]



Die Nummern ① ~ ⑥ geben die Reihenfolge der Berechnungen an.

Um die Berechnungen gemäß der Vorrangordnung bzw. der Verwendung von Klammern ausführen zu können, ist der **PC-1401/1402** mit zusätzlichen Speichern ausgestattet. Diese ermöglichen es, daß mathematische Formeln, die bis zu acht Berechnungsebenen erforderlich machen, direkt eingegeben und berechnet werden können.

Die Klammerfunktion, welche ebenfalls eine Berechnungsebene ergibt, kann, solange keine weiteren Ebenen notwendig sind, bis zu fünfzehnmal verwendet werden.

$a \times (((b - c \times (((d + e) \times f) \div g$ .....

Bis zu 15 Klammern sind möglich

Funktionen mit nur einer Variablen wie (x^2, $1/x$, n!, →DEG, →DMS, usw.) werden sofort nach dem Drücken der Taste berechnet und benötigen somit keine Ebene bzw. keinen Speicher.

Beispiel: Berechnungsebenen bzw. Zwischenspeicher
(ohne Klammerfunktion)

1 [+] 2 [=] (①)	(3)	1 Ebene
1 [+] 2 [×] 3 [=] (① ②)	(7)	2 Ebenen
1 [+] 2 [×] 3 [y^x] 4 [=] (① ② ③)	(163)	3 Ebenen

1 [+] 2 [×] 3 [y^x] 4 [÷] 5 [=] (33,4) 3 und anschl. 2 Ebenen

Mit Drücken der [y^x] -Taste sind 3 Ebenen erreicht. Nach Drücken der [÷] -Taste werden die Berechnungen "y^x" mit höherer Priorität und "x" mit gleicher Priorität als "/" ausgeführt. Es bleiben somit nach Drücken der [÷] -Taste noch 2 Ebenen erhalten. Nach Drücken der [=] -Taste wird zuerst "/" (höhere Priorität) und anschließend "+" ausgeführt.

Beispiel: Berechnungsebenen bzw. Zwischenspeicher (mit Klammerfunktion)

1 [+] 2 [×] 3 [y^x] [(] 4 [÷] 5 [)] [=]

(5,81644937) 4 Ebenen

Die Berechnung in der Klammer wird zuerst ausgeführt. Anschließend folgt "y^x", dann "x" und zuletzt "+".

1 [+] 2 [×] [(] 3 [−] 4 [÷] 5 [)] [=]

(5,4) 4 und anschl. 3 Ebenen

Nach Drücken der [(] -Taste wird zuerst "/", dann "-" ausgeführt. Auf das Drücken der [=] -Taste folgt "x" und zuletzt "+".

2.1.3 WISSENSCHAFTLICHE BERECHNUNGEN

Bei der Durchführung von Berechnungen, die Winkelwerte enthalten, muß auf die Winkeleinheit, die verwendet wird, geachtet werden. Das Vorwählen der Winkeleinheit erfolgt durch Drücken der Tasten [SHIFT] und [DRG] . Die gewählte Winkeleinheit wird im Display angezeigt.

2.1.3.1 Trigonometrische und inverse trigonometrische Funktionen (Arcusfunktionen)

Aufgabe: sin 30° + cos 40°
Tastenfolge: Winkeleinheit auf DEG
30 [sin] + 40 [cos] =
Ergebnis: 1,266044443

Aufgabe: cos 0,25π
Tastenfolge: Winkeleinheit auf RAD
.25 [×] [SHIFT] [π] [=] [cos]
Ergebnis: 0,707106781

Aufgabe: arcsin 0,5
Tastenfolge: Winkeleinheit auf DEG
.5 [SHIFT] [sin⁻¹]
Ergebnis: 30

Aufgabe: arccos -1
Tastenfolge: Winkeleinheit auf RAD
1 [+/-] [SHIFT] [cos⁻¹]
Ergebnis: 3,141592654 (Wert π)

Das Ergebnis einer inversen trigonometrischen Funktion kann nur in folgendem Bereich liegen:

$\theta = \sin^{-1} x,\ \theta = \tan^{-1} x$	$\theta = \cos^{-1} x$
DEG: $-90 \leqq \theta \leqq 90$ [°]	DEG: $0 \leqq \theta \leqq 180$ [°]
RAD: $-\frac{\pi}{2} \leqq \theta \leqq \frac{\pi}{2}$ [rad]	RAD: $0 \leqq \theta \leqq \pi$ [rad]
GRAD: $-100 \leqq \theta \leqq 100$ [g]	GRAD: $0 \leqq \theta \leqq 200$ [g]

2.1.3.2 Hyperbel- und inverse Hyperbelfunktionen (Areafunktion)

Die Anwahl der Hyperbelfunktion wird im Display durch das Symbol "HYP" bzw. bei inverser Hyperbelfunktion durch "SHIFT HYP" angezeigt.

Aufgabe: sinh 4
Tastenfolge:
4 [hyp] [sin]
Ergebnis: 27,2899172

Aufgabe: arsinh 9
Tastenfolge:
9 [SHIFT] [hyp] [sin]
Ergebnis: 2,893443986

2.1.3.3 Logarithmische Funktionen

Aufgabe: ln 21, log 173
Tastenfolge: Natürlicher Logarithmus 21 [ln]
Ergebnis: 3,044522438
Allgemeiner Logarithmus 173 [log]
Ergebnis: 2,238046103

2.1.3.4 Exponentialfunktionen

Aufgabe: $e^{3.0445}$ $10^{2.238}$
Tastenfolge: Exponentialfunktion (e) 3.0445 [SHIFT] [e^x]
Ergebnis: 20,99952881 (siehe ln 21)
Allgem. Exponentialfunktion 2.238 [SHIFT] [10^x]
Ergebnis: 172,9816359 (siehe log 173)

2.1.3.5 Fakultät

Aufgabe: 69! (n! = 1 x 2 x 3 x ...n)
Tastenfolge: 69 [SHIFT] [n!]
Ergebnis: 1,711224524E 98 ($1{,}711224524 \times 10^{98}$)
Anmerkung: Bei Berechnungen der Fakultät kann es sehr leicht zur Überschreitung der Rechenkapazität des **PC-1401/1402** kommen, was die Fehlermeldung "E" zur Folge hat.

Aufgabe: ${}_8P_3 = \frac{8!}{(8-3)!}$

Tastenfolge: 8 [SHIFT] [n!] [÷] [(] 8 [−] 3 [)] [SHIFT] [n!] [=]
Ergebnis: 336

2.1.4 STATISTISCHE BERECHNUNGEN

Die Betriebsart "Statistische Berechnungen" wird durch Drücken der Tasten [SHIFT] (gelb) und [STAT] (unter der roten [C·CE] -Taste) angewählt. Falls sich der Rechner in Betriebsart "RUN" oder "PRO" befindet, muß zuerst in die Betriebsart "CAL" umgeschaltet werden.
Die Markierung (-) im rechten unteren Bereich des Anzeigefelds über der Beschriftung "STAT" gibt an, daß sich der Rechner in Betriebsart "Statistische Berechnungen" befindet.
Das Ausschalten der Betriebsart erfolgt durch Drücken der Tasten [SHIFT] und [STAT].

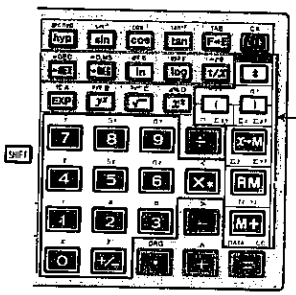
Betriebsart "Statistische Berechnungen" ist angewählt

Wenn "STAT" angewahlt ist, können die folgenden Berechnungen nicht durchgeführt werden:

- Speicherrechnungen
- Rechnungen, die Klammerausdrücke enthalten
- Koordinatenumwandlung
- Umwandlung "Hexadezimal-Dezimal"
- Rechnen mit Zahlen in hexadezimaler Schreibweise

In der Betriebsart "STAT" werden die in der folgenden Abbildung gekennzeichneten Tasten verwendet. Die Zweit-Funktion der verwendeten Tasten wird ebenfalls durch Drücken der [SHIFT]-Taste (gelb) angesprochen.

Tastenfeld für die Betriebsart "Statistische Berechnungen"



Speicherung von Zwischen- und Endergebnissen
Bei der Durchführung von statistischen Berechnungen werden die nachfolgend aufgeführten Ergebnisse automatisch den angegebenen, in der Betriebsart "BASIC" verwendeten Standardvariablen zugeordnet. Die Werte bleiben bei der Umschaltung in die Betriebsart "BASIC" erhalten und können somit direkt für weitere Berechnungen verwendet oder in ein Programm übernommen werden.

Variable	Z	Y	X	W	V	U
Statistik	n	Σx	Σx^2	Σxy	Σy	Σy^2

In diesem Zusammenhang ist jedoch folgendes zu beachten:

- **Den Standardvariablen U bis Z wird ein Wert zugewiesen, d.h., ein gespeichertes Programm kann ggf. beeinträchtigt werden.**

- **Um vorher eingegebene bzw. berechnete Statistikwerte zu löschen, muß, bevor mit einer neuen Berechnung begonnen wird, die Betriebsart "STAT" rückgesetzt und anschließend wieder angewählt werden.** Die Korrektur von falsch eingegebenen Daten ist über die Tasten [SHIFT] [CD] möglich.

Falls mit den errechneten Statistikwerten weitere Berechnungen durchzuführen sind, sollte dies zweckmäßigerweise in der Betriebsart "RUN"

erfolgen. Aus dem folgenden Beispiel ist die grundsätzliche Vorgehensweise ersichtlich. Das Rechnen in Betriebsart "RUN" ist in Abschnitt 2.3.1 ausführlich erläutert.

Beispiel: Berechnung der Summe der Quadrate (S^2) der folgenden vier Werte 205, 221, 226 und 220
Die Berechnungsformel für S^2 lautet:

$$S^2 = \Sigma(x-\bar{x})^2$$
$$= \Sigma x^2 - n\bar{x}^2$$
$$= \Sigma x^2 - \frac{1}{n}(\Sigma x)^2$$

Die Werte für Σx^2 (X), n (Z) und Σx (Y) werden in Betriebsart "STAT" ermittelt. Nach Umschaltung in "RUN" kann anschließend aus diesen Werten die Summe der Quadrate (S^2) errechnet werden.

Tastenfolge:
Umschalten auf "STAT"
[CAL] [SHIFT] [STAT]

Anzeige:

```
                    0.
```

Dateneingabe

205 [DATA] 221 [DATA]

Anzeige:

```
                    4.
```

226 [DATA] 220 [DATA]

Umschalten auf "RUN" und Berechnung von S^2

[BASIC]

Anzeige:

```
>
```

[X] [-] [Y] [X*] [Y] [÷/] [Z]

```
X-Y*Y/Z_
```

[ENTER]

```
                  246.
```

Statistische Berechnungen mit einer Variablen

In den statistischen Berechnungen finden folgende Benennungen Verwendung:

n	Anzahl der eingegebenen Daten
Σx	Gesamtsumme der eingegebenen Daten
Σx^2	Quadratsumme der eingegebenen Daten
$\bar{x}$	Mittelwert der eingegebenen Daten $\bar{x} = \frac{\Sigma x}{n}$

S_x Standardabweichung mit Gesamtheitsparameter " n-1 " (Stichproben-Standardabweichung)

$$S_x = \sqrt{\frac{\Sigma x^2 - n\bar{x}^2}{n-1}}$$

(Wird dann verwendet, wenn die eingegebenen Daten eine Auswahl bzw. Stichprobe aus der Gesamtheit darstellen.)

σ_x Standardabweichung mit Gesamtheitsparameter " n " (Standardabweichung der Gesamtheit)

$$\sigma_x = \sqrt{\frac{\Sigma x^2 - n\bar{x}^2}{n}}$$

(Wird dann verwendet, wenn die eingegebenen Daten als Gesamtheit aufzufassen sind bzw. die Stichproben die Gesamtheit ergeben.)

Bei statistischen Berechnungen mit einer Variablen werden die Daten wie folgt eingegeben:

Wert [DATA] oder, falls mehrmals der gleiche Wert einzugeben ist,

Wert [×] Haufigkeit [DATA]

Aufgabe: Berechnen der Standardabweichung Sx, des Mittelwerts x und der Varianz $(Sx)^2$ der folgenden Daten:

Wert	35	45	55	65
Häufigkeit	1	1	5	2

Nach jeder Eingabe wird im Display die Anzahl der bisher eingegebenen Daten (n) angezeigt.

Tastenfolge:		Anzeige:
35	[DATA]	1.
45	[DATA]	2.
55 x 5	[DATA]	7.
65 x 2	[DATA]	9.
[SHIFT] [TAB] [2]		(Einst. auf zwei Nachkommastellen)
Mittelwert	[SHIFT] [$\bar{x}$]	53.89
Standardabweichung	[SHIFT] [S_x]	9.28
Varianz	[$\bar{x}^2$]	86.11

Korrektur eingegebener Daten (CD): Die zuletzt eingegebenen Werte 65 x 2 sind falsch und müssen auf 60 x 2 berichtigt werden.

	Tastenfolge:	Anzeige:
	65 [×] 2 [SHIFT] [CD]	7.00
	60 [×] 2 [DATA]	9.00
Mittelwert	[SHIFT] [$\bar{x}$]	52.78
Standardabweichung	[SHIFT] [S_x]	7.95
Varianz	[x^2]	63.19

Statistische Berechnungen mit zwei Variablen und Lineare Regression

Zusätzlich dazu, daß die statistischen Funktionen für zwei Variable (x und y) vorhanden sind und über die [Σxy]-Taste die Summe der xy-Werte errechnet werden kann, ist es in den statistischen Berechnungen für zwei Variable möglich, eine Beziehung (Korrelation) zwischen den beiden Datenfolgen herzustellen.

Jedes eingegebene Datenpaar besteht aus einem x- und y-Wert. Aus den Datenpaaren läßt sich eine Regressionslinie ableiten. Da die Beziehung zwischen den Datenpaaren über eine gerade Linie hergestellt wird, spricht man von einer "Linearen Regression".

Die Lineare Regression weist drei wichtige Werte auf: r, a und b. Die Gleichung der geraden Linie ist $y = a + bx$, wobei a den Punkt beschreibt, an dem die Linie die y-Achse schneidet, während b die Steilheit angibt.

Der Korrelationskoeffizient r ermöglicht eine Aussage darüber, mit welcher Genauigkeit ein Bezug zwischen den beiden Datenfolgen hergestellt werden kann. Eine perfekte Korrelation zwischen zwei Werten ist bei $r = 1$ bzw. $r = -1$ (negative Korrelation) gegeben, d.h., daß in diesem Fall, wenn der Wert der einen Variablen bekannt ist, der andere Wert mit einer Genauigkeit von 100 % ermittelt werden kann. Je weiter der Korrelationskoeffizient r von 1 entfernt ist, um so ungenauer sind die ermittelten Werte.

Die nachfolgende Tabelle soll eine Bewertung des Korrelationskoeffizienten r ermöglichen.

	Korrelationskoeffizient (r)	Genauigkeit
Positive Korrelation	+ 0,8 ... + 1,0	sehr hoch
	+ 0,6 ... + 0,8	hoch
	+ 0,4 ... + 0,6	mittelmäßig
	+ 0,2 ... + 0,4	gering
	- 0,2 ... + 0,2	----------
	- 0,2 ... - 0,4	gering
Negative Korrelation	- 0,4 ... - 0,6	mittelmäßig
	- 0,6 ... - 0,8	hoch
	- 0,8 ... - 1,0	sehr hoch

Berechnungsformeln:

$$r = \frac{S_{xy}}{\sqrt{S_{xx} \cdot S_{yy}}}$$

$$\left[\begin{array}{l} S_{xx} = \Sigma x^2 - \dfrac{(\Sigma x)^2}{n} \\ S_{yy} = \Sigma y^2 - \dfrac{(\Sigma y)^2}{n} \\ S_{xy} = \Sigma xy - \dfrac{\Sigma x \cdot \Sigma y}{n} \end{array}\right]$$

$$\left.\begin{array}{l} a = \bar{y} - b\bar{x} \\ b = \dfrac{S_{xy}}{S_{xx}} \end{array}\right\} \quad y = a + bx$$

Beispiel 1 :
Ermittlung der voraussichtlichen Punktezahl in der Englischprüfung aus der in der Mathematikprüfung erreichten Punktezahl.
Die Grundlage der statistischen Berechnung bilden die erreichten Punktezahlen von sechs zufällig ausgewählten Prüfungskandidaten.

Prüfungskandidat Nr. n	Punkte in Math. x	Punkte in Englisch y
1	82	79
2	53	50
3	61	87
4	74	96
5	51	73
6	51	73

Tastenfolge:	Anzeige:
82 [x,y] 79 [DATA]	1.
53 [x,y] 50 [DATA]	2.
61 [x,y] 87 [DATA]	3.

Tastenfolge:	Anzeige:
74 [x,y] 96 [DATA]	4.
51 [x,y] 73 [X] 2 [DATA]	6.

Anmerkung: Mehrere identische Werte für x, y können über die [X] - Taste eingegeben werden.

Tastenfolge:	Anzeige:
[SHIFT] [TAB] [2]	
[SHIFT] [r]	0.57
[SHIFT] [a]	34.26
[SHIFT] [b]	0.68

Der Korrelationskoeffizient r = 0,57 besagt, daß eine mittelmäßige Genauigkeit zu erwarten ist.
Die Gleichung der Geraden lautet für dieses Beispiel
y = 34,62 + 0,68x.

Tastenfolge:	Anzeige:
90 [SHIFT] [y']	95.33

Aufgrund dieser statistischen Berechnung kann ein Prüfungskandidat, der in der Mathematikprüfung 90 Punkte erreichte, damit rechnen, in der Englischprüfung 95 Punkte zu erreichen.

Beispiel 2 :
Kann aus dem Gewicht, das ein Mann im Alter von 65 Jahren besitzt, auf die Gesamtlebensdauer geschlossen werden? Im Jahre 1950 wurden zehn Männer von annähernd gleicher Körpergröße (1,80 m) für ein Experiment ausgewählt, das Aufschluß darüber geben sollte, ob das Gewicht einen Einfluß auf die Gesamtlebensdauer hat.

Nummer	1	2	3	4	5	6	7	8	9	10
Lebensdauer	72	67	69	85	91	68	77	74	70	82
Gewicht mit 65	83	102	91	77	77	88	79	79	90	78

Tastenfolge: "STAT", [SHIFT] [TAB] [•]
83 [x,y] 72 [DATA]
102 [x,y] 67 [DATA]
(alle Daten eingeben)

[SHIFT] [r] -0.787476214

Der Korrelationsfaktor r gibt eine relativ hohe negative Korrelation an. Ein höheres Gewicht bedeutet somit eine kürzere Lebensdauer.

Zum Zeichnen der Regressionslinie sind die Werte für a und b erforderlich.

SHIFT [a]	140.977975 (y-Achse)
SHIFT [b]	-0.775805391 (Steilheit)

Wie alt wird ein ca. 1,80 m großer Mann, der 86 kg wiegt?

86 SHIFT [x'] 70.86567797 Jahre

Um 90 Jahre alt zu werden, sollte ein 1,80 m großer Mann wie schwer sein?

90 SHIFT [y'] 71.15548981 kg

Wie schwer müßte man sein, um 150 Jahre alt zu werden? Aus dieser letzten Frage wird deutlich, daß die Lineare Regression nur in einem bestimmten eingegrenzten Bereich brauchbare Ergebnisse liefert.

2.1.5 RECHENBEREICH

Arithmetische Berechnungen:

Erster Operand
Zweiter Operand
Rechenergebniss
} $\pm 1 \times 10^{-99}$ bis $\pm 9{,}999999999 \times 10^{99}$ und 0

Wissenschaftliche Berechnungen:

Funktion	Rechenbereich	Anmerkung
$\sin x$ $\cos x$ $\tan x$	DEG: $\|x\| < 1 \times 10^{10}$ RAD: $\|x\| < \frac{\pi}{180} \times 10^{10}$ GRAD: $\|x\| < \frac{10}{9} \times 10^{10}$ Für tan x gelten folgende Einschränkungen DEG: $\|x\| = 90(2n-1)$ RAD: $\|x\| = \frac{\pi}{2}(2n-1)$ n = Ganze Zahl GRAD: $\|x\| = 100(2n-1)$	
$\sin^{-1} x$ $\cos^{-1} x$	$-1 \leqq x \leqq 1$	
$\tan^{-1} x$	$\|x\| < 1 \times 10^{100}$	
$\ln x$ $\log x$	$1 \times 10^{-99} \leqq x < 1 \times 10^{100}$	($\ln x = \log_e x$)

Funktion		Rechenbereich	Anmerkung
e^x		$-1 \times 10^{100} < x \leq 230.2585092$	(e ≒ 2.718281828)
10^x		$-1 \times 10^{100} < x < 100$	
y^x		• $y > 0$: $-1 \times 10^{100} < x \log y < 100$ • $y = 0$: $x > 0$ • $y < 0$: x: ganzzahlig oder 1/x: ungerade $-1 \times 10^{100} < x \log \lvert y \rvert < 100$	$y^x = 10^{x \cdot \log y}$
$\sqrt[x]{y}$		• $y > 0$: $-1 \times 10^{100} < \frac{1}{x} \log y < 100$, $x \neq 0$ • $y = 0$: $x > 0$ • $y < 0$: x oder 1/x: ganzzahlig $(x \neq 0)$ $-1 \times 10^{100} < \frac{1}{x} \log \lvert y \rvert < 100$	$\sqrt[x]{y} = 10^{\frac{1}{x} \cdot \log y}$
$\sqrt[3]{x}$		$\lvert x \rvert < 1 \times 10^{100}$	
sinh x cosh x tanh x		$-227.9559242 \leq x \leq 230.2585092$	
$\sinh^{-1} x$		$\lvert x \rvert < 1 \times 10^{50}$	
$\cosh^{-1} x$		$1 \leq x < 1 \times 10^{50}$	
$\tanh^{-1} x$		$\lvert x \rvert < 1$	
$\sqrt{x}$		$0 \leq x < 1 \times 10^{100}$	
x^2		$\lvert x \rvert < 1 \times 10^{50}$	
$\frac{1}{x}$		$\lvert x \rvert < 1 \times 10^{100}$ $x \neq 0$	
n!		$0 \leq n \leq 69$ (n = Ganze Zahl)	
→DEG		$\lvert x \rvert < 1 \times 10^{100}$	
→ D.MS		$\lvert x \rvert < 1 \times 10^{100}$	
HEX → DEC		0 ≦ x ≦ 2540BE3FF FDABF41C01 ≦ x ≦ FFFFFFFFFF	x ist in "HEX" eine ganze Zahl
DEC → HEX		$\lvert x \rvert \leq 9999999999$	x = ganzzahlig
$x, y \rightarrow r, \theta$		$(x^2 + y^2) < 1 \times 10^{100}$ $\frac{y}{x} < 1 \times 10^{100}$	$r = \sqrt{x^2 + y^2}$ $\theta = \tan^{-1} \frac{y}{x}$
$r, \theta \rightarrow x, y$		$r < 1 \times 10^{100}$, $\lvert r \sin\theta \rvert < 1 \times 10^{100}$ $\lvert r \cos\theta \rvert < 1 \times 10^{100}$	$x = r \cos\theta$ $y = r \sin\theta$ θ ist in derselben Bedingung wie x von sin x, cos x
Statistische Berechnungen	Data	$\lvert x \rvert < 1 \times 10^{50}$	
	CD	$\lvert y \rvert < 1 \times 10^{50}$ $\lvert \Sigma x \rvert < 1 \times 10^{100}$ $\Sigma x^2 < 1 \times 10^{100}$ $\lvert \Sigma y \rvert < 1 \times 10^{100}$ $\Sigma y^2 < 1 \times 10^{100}$ $\lvert \Sigma xy \rvert < 1 \times 10^{100}$ $\lvert n \rvert < 1 \times 10^{100}$	

(Forsetzung der Tabelle)

Funktion		Rechenbereich	Anmerkung
Statistische Berechnungen	$\bar{x}$	$n \neq 0$	
	Sx	$n \neq 1$ $0 \leqq \frac{\Sigma x^2 - n\bar{x}^2}{n-1} < 1 \times 10^{100}$	
	σx	$n \neq 0$ $0 \leqq \frac{\Sigma x^2 - n\bar{x}^2}{n} < 1 \times 10^{100}$	
	$\bar{y}$	$n \neq 0$	
	Sy	$n \neq 1$ $0 \leqq \frac{\Sigma y^2 - n\bar{y}^2}{n-1} < 1 \times 10^{100}$	
	σy	$n \neq 0$ $0 \leqq \frac{\Sigma y^2 - n\bar{y}^2}{n} < 1 \times 10^{100}$	
	r	$n \neq 0$ $0 < \lvert (\Sigma x^2 - n\bar{x}^2) \cdot (\Sigma y^2 - n\bar{y}^2) \rvert < 1 \times 10^{100}$ $\left\lvert \Sigma xy - \frac{\Sigma x \cdot \Sigma y}{n} \right\rvert < 1 \times 10^{100}$ $\left\lvert \frac{\Sigma xy - \frac{\Sigma x \cdot \Sigma y}{n}}{\sqrt{(\Sigma x^2 - n\bar{x}^2) \cdot (\Sigma y^2 - n\bar{y}^2)}} \right\rvert < 1 \times 10^{100}$	
	b	$n \neq 0$ $0 < \lvert \Sigma x^2 - n\bar{x}^2 \rvert < 1 \times 10^{100}$ $\left\lvert \Sigma xy - \frac{\Sigma x \cdot \Sigma y}{n} \right\rvert < 1 \times 10^{100}$ $\left\lvert \frac{\Sigma xy - \frac{\Sigma x \cdot \Sigma y}{n}}{\Sigma x^2 - n\bar{x}^2} \right\rvert < 1 \times 10^{100}$	
	a	a ist in derselben Bedingung wie b, und $\lvert \bar{y} - b\bar{x} \rvert < 1 \times 10^{100}$	
	y'	$\lvert a + bx \rvert < 1 \times 10^{100}$	
	x'	$\left\lvert \frac{y-a}{b} \right\rvert < 1 \times 10^{100}$	

Die Genauigkeit beträgt in der Regel für die Fließkomma-Schreibweise ±1 in der 10. Stelle und für die wissenschaftliche Schreibweise ±1 in der 9. Nachkommastelle der Mantisse.
Die Berechnungsgenauigkeit sinkt jedoch in der Nähe des Singulär-

Punktes und des Umkehrpunktes der Funktion ab.

Ferner ist zu beachten, daß bei Kettenrechnungen eine Kummulierung des Fehlers erfolgt und damit mit jedem Rechengang eine Verschlechterung der Genauigkeit einhergeht. Der gleiche Effekt tritt rechnerintern bei der Durchführung von Funktionen wie y^x und $\sqrt[x]{y}$ auf.

2.2 SHARP PC-1401/1402 - EINSATZ ALS BASIC-RECHNER

2.2.1 RECHNEN OHNE PROGRAMMUNTERSTÜTZUNG (RUN-MODE)

Der **PC-1401/1402** kann auch in der Programmiersprache BASIC programmiert werden. Dabei kann er auf zwei verschiedene Arten verwendet werden. Zum einen können Sie Ihre Berechnungen innerhalb eines Programms abarbeiten, zum anderen können die Anweisungen und Funktionen direkt eingegeben und ausfeführt werden (RUN-Mode).

Dabei kann man den **PC-1401/1402** wie einen einfachen Taschenrechner benutzen. Um das Ergebnis einer Rechnung zu erhalten muß man anstelle der '='-Taste die [ENTER] -Taste drücken.

Die allgemeine Form einer Rechnung im RUN-Mode ist:

[C·CE] numerischer Ausdruck [ENTER]

Als 'numerischer Ausdruck' wird jede mathematische Formel bezeichnet, die einen Zahlenwert als Ergebnis hat.

2.2.1.1 Grundrechnungsarten

a) Addition

[5] [Ø] [+] [5] [Ø] [ENTER]

```
100.
```

b) Subtraktion

[1] [Ø] [Ø] [-] [5] [Ø] [ENTER]

```
50.
```

c) Division

[3] [Ø] [Ø] [/] [5] [ENTER]

```
60.
```

d) Multiplikation

[6] [Ø] [*] [1] [Ø] [ENTER]

```
600.
```

e) Klammerrechnungen, Exponentation

[(] [6] [7] [5] [+] [6]
[7] [5] [Ø] [)] [/] [4]
[5] [Ø] [Ø] [Ø] [ENTER]

```
Ø.165
```

Hinweise:

1. Erst nach Drücken der [ENTER] Taste wird die Rechnung oder das Kommando ausgeführt und das Ergebnis angezeigt.

2. Das Ergebnis der Rechnung kann für weitere Berechnungen verwendet werden. Dazu die Zahl nicht mit der [CA] [C·CE] -Taste löschen, sondern das nachste Operationszeichen der neuen Formel eingeben (+ - * / ^).

Beispiel:

[3] [0] [.] [0] [*] [1] [5] [0] [ENTER]	45000.
[*] [.] [1] [5]	45000.*.15_
[ENTER]	6750.
[-] [4] [0] [0] [0]	6750.-4000_
[ENTER]	2750.
[-] [1] [2] [2] [5] [ENTER]	1525.
[-] [2] [2] [0] [0] [ENTER]	-675.

2.2.1.2 Rechnengenauigkeit

Der **PC-1401/1402** verarbeitet Zahlen mit einer Genauigkeit bis zu 10 Stellen. Dies führt bei einigen Rechnungen zu Rundungsfehlern. Die Genauigkeitsbeschränkung tritt auch bei der Exponentation oder beim Addieren von Zahlen sehr unterschiedlicher Größe auf.

Beispiele:

Eingabe

```
A=199^3
```

ENTER

Ausgabe

```
7880598.999
```

Eingabe

```
A-199*199*199
```

ENTER

Ausgabe

```
-0.001
```

Eingabe

```
1E12+1-1E12
```

ENTER

Ausgabe

```
0.
```

2.2.1.3 Wissenschaftliche Schreibweise (Notation)

Man kann Zahlen in den **PC-1401/1402** auch im wissenschaftlichen Format eingeben. Damit ist es möglich, Zahlen bis zu einer Größenordung von ±9.999999999E ± 99 einzugeben und zu verarbeiten.

Wird dieser Wertbereich überschritten, erscheint die Fehlermeldung ERROR 2.

Ist der Betrag einer Zahl kleiner als 1 E-99, so wird die Zahl auf Null gesetzt.

Wird eine Zahl mit einem mehr als zweistelligen Exponenten eingegeben, so werden nur die beiden zuletzt eingegebenen Ziffern bewertet.

Hinweis:

1. Wie im englischen Sprachraum üblich, wird nicht das Komma, sondern der Punkt zur Trennung von ganzem und gebrochenem Anteil von Zahlen verwendet.

2. Für den Operator "geteilt durch" wird ein Schrägstrich (/) anstelle des Doppelpunkts gesetzt.

3. Für "multipliziert mit" muß ein Stern (*) gesetzt werden.

4. Als Zeichen für die Exponentation wird ein Dach (^) verwendet. Ist die Basis negativ, muß sie in Klammern gesetzt werden. Als Exponenten sind bei negativen Grundzahlen nur ganze Zahlen zulässig. Ansonsten erscheint die Fehlermeldung ERROR 2.

5. Sind mehr als 10 Ziffern für die Darstellung eines Ergebnisses erforderlich, wird es im wissenschaftlichen Format angezeigt. Das Anzeigeformat kann durch die BASIC-Anweisung USING verändert werden.

2.2.1.4 Editier-/Korrekturmöglichkeit

Tippfehler können beim **PC-1401/1402** während der Eingabe oder bei der Überprüfung der Eingaben korrigiert werden.

Mit den Cursor-Tasten [◄] und [►] können Sie den Cursor nach links oder rechts bewegen, ohne daß dabei die eingegebe Formel verändert wird. Zur Korrektur stellen Sie den Cursor über das zu korrigierende Zeichen und überschreiben dieses mit dem neuen Zeichen.

Zum Löschen eines Zeichens den Cursor über das zu löschende Zeichen stellen und die Tasten [SHIFT] und [DEL] drücken.

Zum Einfügen von Zeichen drücken Sie die Tasten [SHIFT] und [INS] . Es wird ein Platzhalter (_) in die Formel eingefügt, der durch ein neues Zeichen überschrieben werden kann. Durch Drücken der [ENTER]-Taste werden alle überflüssigen Platzhalter aus der Formel gelöscht.

Ist das Ergebnis einer Berechnung schon ausgegeben oder erschien bei der Berechnung eine Fehlermeldung, so kann die Ausgangsformel mit den Tasten [↑] oder [↓] wieder zur Anzeige gebracht und wie oben beschrieben geändert werden.

Hinweis:

BASIC-Schlüsselwörter wie LET, SIN, COS, PRINT usw. werden nach Betätigung der [ENTER]-Taste intern abgekürzt. Obwohl das Schlüsselwort unverändert auf der Anzeige erscheint, wird es beim Editieren durch ein einziges neues Zeichen vollständig überschrieben. Will man sehr lange Programmzeilen eingeben, kommt man vorerst nur bis zum 80. Zeichen. Drücken Sie nun die [ENTER]-Taste und stellen Sie den Cursor an das Zeilenende. Nun können Sie noch zusätzlich einige Zeichen eingeben. Die Eingabe muß wieder mit [ENTER] abgeschlossen werden.

2.2.1.5 Mathematische Funktionen/Klammerregeln

Der **PC-1401/1402** verfügt über alle gebräuchlichen mathematischen Funktionen. Diese können, im Gegensatz zu seinen Vorgängern, bei denen die Eingabe nur über die Buchstabentasten möglich war, über Funktionstasten eingegeben und im BASIC verarbeitet werden.

Beispiel:

[sin] [Ø] [ENTER]

```
                Ø.
```

Die Reihenfolge der Einzelschritte bei der Berechnung eines komplexen mathematischen Ausdrucks wird durch Klammern festgelegt.

Wie in der Mathematik gibt es dabei eine implizite Klammerung, d.h., daß bestimmte Operationen vor anderen den Vorrang haben. Der **PC-1401/1402** verfügt über 15 Klammerebenen.

Beispiele:

3 * 5 + 7 * 4

ist gleichbedeutend mit

(3 * 5) + (7 * 4)

Die Rangfolge der mathematischen Operatoren ist:

1. Klammern
2. Abruf von PI, MEM, Variablen
3. Funktionsoperationen in Bezug auf das folgende Argument
4. Exponentiation
5. Vorzeichen +, −
6. Multiplikation, Division
7. Addition, Subtraktion
8. Vergleichsoperationen
9. Logische Operationen

2.2.1.6 Hexadezimal-(Sedezimal-)Zahlen

Natürliche Zahlen im Bereich zwischen 0 und 65535 können in den **PC-1401/1402** auch in sedezimaler Form eingegeben werden.
Der Sedezimalzahl wird dabei ein &-Zeichen vorangestellt. Zur hexadezimalen Darstellung einer Zahl dienen die Ziffern zwischen 0 bis 9 und die Buchstaben A bis F.

Beispiel:

```
& A                                   ENTER
```

Ausgabe

```
                                  10.
```

Eingabe

```
& F F F F                             ENTER
```

Ausgabe

```
                               65535.
```

Die Ausgabe von Zahlen in Sedezimalform erfolgt beim PC-1401/1402 mit der Tastenfolge:

C·CE Dezimalzahl ·HEX ENTER

Beispiel:

```
123  ·HEX                        7B.
```

Hinweise:

1. In folgenden Fällen läßt sich eine Umwandlung nicht durchführen:
   a) Wenn die Zahl eine Dezimalstelle hat, die bei Fließkommabetrieb nicht 0 ist.
   b) Wenn eine Zahl in wissenschaftlicher Schreibweise dargestellt ist.

2. Die HEX-Funktion dient nur zur Ausgabe/Anzeige einer Dezimalzahl in sedezimaler Form. Das Ergebnis der Umwandlung kann nicht mehr weiterverarbeitet werden.

2.2.1.7 Textausdrücke

Textausdrücke sind Bestandteil der BASIC-Sprache. Sie können im RUN-Mode ohne Programmunterstützung eingegeben werden.

Man unterscheidet Textkonstanten, Textvariablen, Textfunktionen und zusammengesetzte Texte. Eine Textkonstante ist eine beliebige Zeichenfolge, die durch Anführungszeichen begrenzt ist, wobei die Anführungszeichen nicht Bestandteil der Textkonstante sind.

Beispiel:

[C·CE] Eingabe

```
"SHARP"
```

[ENTER]

Ausgabe

```
SHARP
```

Textvariablen können bis zu sieben Zeichen enthalten. Über die DIM-Anweisung lassen sich Textvariablen bis zu 80 Zeichen erzeugen.

Beispiel:

Eingabe

```
A$ = "SHARP"
```

[ENTER]

Ausgabe

```
SHARP
```

Texte können mit Hilfe des "+"-Zeichens aneinandergefügt werden.

Beispiel:

Eingabe

```
A$ = "POCKET"
```

[ENTER]

Ausgabe

```
POCKET .
```

Eingabe

A$ + "␣COMPUTER"	ENTER

Ausgabe

POCKET COMPUTER

Hinweis:

Auch das Leerzeichen (Space) innerhalb eines Textes ist ein gültiges Zeichen.

2.2.1.8 Logische Vergleichsausdrücke

Der **PC-1401/1402** kennt folgende Vergleichsausdrücke:

Mathematisches Symbol	BASIC	Deutsche Sprechweise
<	<	kleiner als
≤	<=	kleiner gleich
=	=	gleich
≥	>=	größer gleich
>	>	größer als
≠	<>	ungleich

Mit diesen Operatoren können sowohl zwei numerische als auch zwei Textausdrücke miteinander verglichen werden. Ist die Vergleichsaussage richtig, liefert der Rechner das Ergebnis "1", ist sie falsch, ist das Ergebnis "0". Im Zusammenhang mit der BASIC-Anweisung IF erhalten diese Vergleichsoperatoren eine besondere Bedeutung.

Beispiele:

Eingabe

C·CE	2 < 1	ENTER

Ausgabe

0.

b) Eingabe

[C-CE] `1 <= 2` ENTER

Ausgabe

```
1.
```

c) Eingabe

[C-CE] `SIN1<.7` ENTER

Ausgabe

```
0.
```
(im RAD-Mode)

```
1.
```
(im DEG- und GRAD-Mode)

d) Eingabe

[C-CE] `"WERNER" > "ANTON"` ENTER

Ausgabe

```
1.
```

Die Textausdrücke "WERNER" und "ANTON" werden entsprechend dem ASCII-Code auf die lexikographische Reihenfolge hin überprüft.

Hinweis:

Da das Ergebnis eines logischen Ausdrucks eine Zahl ist, kann dieses als numerische Variable gespeichert und weiterverarbeitet werden.

2.2.2 SPRACHELEMENTE

2.2.2.1 Numerische Konstante

Eine numerische Konstante kann sein:

- eine ganze Zahl (positiv oder negativ)
- eine Dezimalzahl
- eine Zahl in wissenschaftlicher Schreibweise
- eine Sedezimal-(Hexadezimal-)Zahl

Beispiele:

```
513
-2376
11.745
1.23456788E-12
&AA2B
```

2.2.2.2 Textkonstante

Eine Textkonstante ist eine beliebige Zeichenfolge, die durch Anführungszeichen (") begrenzt wird.

Beispiele:

```
"SHARP"
"     "  (Textkonstante enthält Leerzeichen)
""       (Textkonstante der Länge 0)
```

2.2.2.3 Numerische Variable

Unter einer numerischen Variable versteht man den Speicherplatz, der zur Aufnahme des jeweiligen, zugehörigen Zahlenwertes bereitsteht.

Der Variablenname setzt sich aus einem Buchstaben oder einem Buchstaben mit einem in Klammern folgenden Indexwert zusammen. Dabei kann der Index ebenfalls durch eine Variable dargestellt werden.

Beispiele:

```
A
A(27)
B(1)
A(Z)
AB
A1
XY(2,4)
```

2.2.2.4 Textvariable

Unter einer Textvariable versteht man den Speicherplatz, der zur Aufnahme der jeweiligen, zugehörigen Zeichenfolge bereitsteht.

Der Name der Textvariable ist im Prinzip gleich aufgebaut wie der der numerischen Variable mit einem zusätzlichen "$"-Zeichen hinter dem Buchstaben.

Beispiele:

A$
B$(2)
C$(Z)

2.2.2.5 Numerische Funktionen, Textfunktionen

Eine numerische Funktion setzt sich aus einem Operator und einem oder mehreren Parametern zusammen. Die Parameter werden hinter den Funktionsnamen gestellt und können je nach Funktion numerische oder Textausdrücke sein. Sind mehrere Parameter erforderlich, so werden diese in Klammern gesetzt und durch Kommata getrennt.

Das Ergebnis einer numerischen Funktion ist ein Zahlenwert, das einer Textfunktion ein Zeichen bzw. eine Zeichenfolge.

Beispiele:

LN60	numerische Funktion; natürlicher Logarithmus des numerischen Parameters 60
ASC"A"	numerische Funktion; Ergebnis ist die numerische Konstante 65, die den ASCII-Code der Textkonstante "A" darstellt.
CHR$65	Textfunktion; Ergebnis ist die Textkonstante "A".
MID$("SHARP",2,2)	Textfunktion; Ergebnis ist die Textkonstante "HA".

Hinweis:

Funktionsoperatoren haben eine höhere Priorität als andere Operatoren.

Beispiel:

SIN A + B = (SIN A) + B

Soll der Sinus der Summe (A + B) gebildet werden, so müssen Klammern gesetzt werden:

SIN (A + B)

2.2.2.6 Numerischer Ausdruck

Ein numerischer Ausdruck setzt sich aus einer numerischen Konstanten, Variablen, numerischen Funktion oder deren Verknüpfung durch die arithmetischen Operatoren +, -, *, /, ^ , AND,OR und NOT und die Zusammenfassung durch Klammern zusammen.

Beispiele:

```
15
(-8)
SIN 45
A + B/C
(C*(A*X+B)*5/(D*C))+10
```

2.2.2.7 Textausdruck

Ein Textausdruck besteht aus einer Textkonstanten, Textvariablen oder eine Textfunktion und deren Verknüpfung durch das "+"-Zeichen. Das Ergebnis ei Textausdrucks ist eine Zeichenfolge.

Beispiele:

```
"A" + STR$ A
A$ + "PC"
```

2.2.2.8 Logische Vergleichsausdrücke

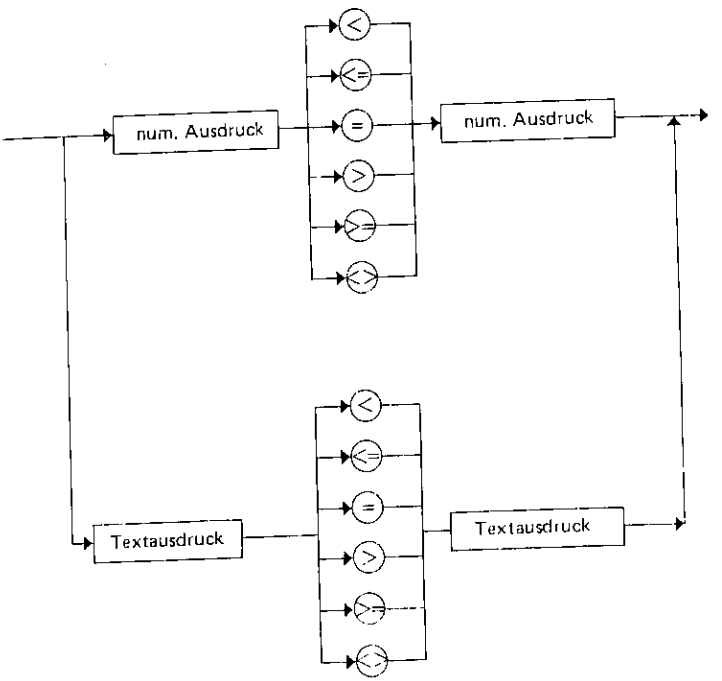
Ein logischer Vergleichsausdruck ist der Vergleich zweier Ausdrücke (numerisch oder Text) durch die Operatoren:

$>, >=, <, <=, <>$

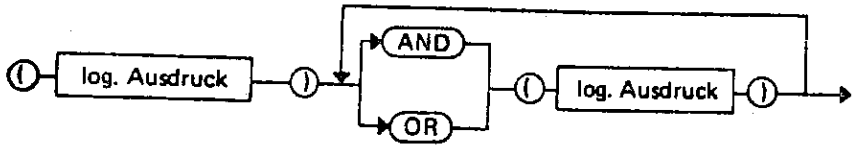
und die Verknüpfung solcher Vergleiche mit den Booleschen Operatoren AND, OR und NOT.

Operation	Mathematisches Symbol	BASIC
Negation	$\neg$	NOT
UND-Funktion	$\wedge$	AND
ODER-Funktion	$\vee$	OR

Syntax:



Logische Vergleichsausdrücke lassen sich wie folgt verknüpfen:



Ist die Vergleichsaussage richtig, so ist das Ergebnis des logischen Ausdrucks "1", ist sie falsch, ist das Ergebnis "0".

Werden Textausdrücke miteinander verglichen, so werden die Textausdrücke auf lexikographische Reihenfolge geprüft; diese richtet sich beim **PC-1401/1402** nach dem ASCII-Code (siehe Anhang).

Werden zwei Textausdrücke mit unterschiedlicher Länge verglichen, so werden beim Vergleich die fehlenden Stellen des kürzeren Ausdrucks mit dem ASCII-Zeichen Null aufgefüllt.

Logische Ausdrücke werden in der BASIC-Anweisung IF verwendet.

Beispiele:

	Ergebnis
1 < 2	1
1 + 2 + 3 < 2 + 3 + 4	1
1 >= 2	0
1 < 2 < 3	1
"ANTON" < "WERNER"	1
"ANTON" > "ANTON 1"	0 ("1" ist größer als ASCII-Null)
(1 < 2) AND (3 < 4)	1
("ANTON" < "WERNER") OR ("ANTON" = "ANTON 1")	1

2.2.3 VARIABLEN IM SHARP PC-1401/1402

Variablen sind Bezeichner für Größen, deren Wert erst im Laufe des Programmablaufs festgelegt werden. Sie werden beim Rechner durch Speicherplätze realisiert, denen man unterschiedliche Daten durch Einlesen oder durch Auswerten eines bestimmten Ausdrucks zuordnen kann. Je nachdem, ob es sich bei der Variablen um einen numerischen oder um einen Textausdruck handelt, nennt man solche Variablen numerische Variablen oder Textvariablen.

Zur Unterscheidung zwischen numerischen und Textvariablen muß jeder Name einer Textvariablen mit dem "$"-Zeichen enden. Beide Typen können als einfache oder indizierte Variable geschrieben werden. Zur leichteren Identifizierung erhalten Sie einen Variablennamen als symbolische Adresse für den Speicherplatz. Mit Variablen lassen sich Gesetzmäßigkeiten unabhängig von ihrem jeweiligen Wert allgemein ausdrücken.

Im **PC-1401/1402** stehen folgende zwei verschiedene Speicherbereiche für Variablen zur Verfügung (s. Abb. Anhang H, S. 229).

a) Standardvariablenspeicher
b) Feldvariablenspeicher

Der Standardvariablenspeicher ist ein begrenzter Speicherbereich und dient ausschließlich zur Aufnahme der 26 Standardvariablen.

Der Feldvariablenspeicher ist Bestandteil des Hauptspeichers, der auch zur Aufnahme von Programmen dient. Die Kapazität des Hauptspeichers kann wahlweise für Programme und für Feldvariablen benutzt werden.

Folgende Variablentypen stehen im **PC-1401/1402** zur Verfügung:

1. Standardvariablen
   a) einfache numerische Variable
   b) einfache Textvariable
   c) indizierte numerische Variable
   d) indizierte Textvariable

2. Feldvariablen
   a) indizierte eindimensionale numerische Feldvariable (Vektor)
   b) indizierte eindimensionale Textfeldvariable (Vektor)
   c) indizierte zweidimensionale numerische Feldvariable (Matrizen)
   d) indizierte zweidimensionale Textfeldvariable (Matrizen)

2.2.3.1 Standardvariable

Für die Standardvariablen stehen 26 reservierte Speicherbereiche zur Verfügung, die wahlweise als numerische oder als Textvariable belegt werden können.

Den Standardvariablen können also wahlweise numerische oder Textwerte zugeordnet werden. Bei der Wertabfrage der Standardvariablen muß der Variablenname dem Inhalt dieser Variablen entsprechen. Wird z.B. eine numerische Variable als Textvariable aufgerufen, und umgekehrt, so wird am Display die Fehlermeldung ERROR 9 angezeigt.

Standardvariable:

A = A$ = A(1) = A$(1)
B = B$ = A(2) = A$(2)
C = C$ = A(3) = A$(3)
D = D$ = A(4) = A$(4)
E = E$ = A(5) = A$(5)
F = F$ = A(6) = A$(6)
G = G$ = A(7) = A$(7)
H = H$ = A(8) = A$(8)
I = I$ = A(9) = A$(9)
J = J$ = A(10) = A$(10)
K = K$ = A(11) = A$(11)
L = L$ = A(12) = A$(12)
M = M$ = A(13) = A$(13)
N = N$ = A(14) = A$(14)
O = O$ = A(15) = A$(15)
P = P$ = A(16) = A$(16)
Q = Q$ = A(17) = A$(17)
R = R$ = A(18) = A$(18)
S = S$ = A(19) = A$(19)
T = T$ = A(20) = A$(20)
U = U$ = A(21) = A$(21)
V = V$ = A(22) = A$(22)
W = W$ = A(23) = A$(23)
X = X$ = A(24) = A$(24)
Y = Y$ = A(25) = A$(25)
Z = Z$ = A(26) = A$(26)

2.2.3.2 Einfache Variablen

Einfache Variablen werden mit den Namen A bis Z als numerische Variable bzw. A$ bis Z$ als Textvariable aufgerufen.

Einer einfachen numerischen Variablen kann man eine Zahl mit maximal 10 Stellen, einen zweistelligen Exponenten und die Vorzeichen zuordnen.

Beispiele:

```
A = 123
A = SIN X
A = B + C
A = B * C
```

Einer Textvariablen kann man Zeichenfolgen bis zu sieben Zeichen, bestehend aus Buchstaben, Zahlen, Sonderzeichen und Leerstellen, zuweisen. Bei der Wertzuweisung muß der Text durch Anführungszeichen begrenzt werden.

Beispiele:

```
B$="TEXT"
B$="T E X T"
```

2.2.3.3 Indizierte Variablen

Indizierte Variablen werden mit den Namen A(1) bis A(26) als numerische bzw. mit A$(1) bis A$(26) als Textvariable aufgerufen.

Die indizierten Standardvariablen sind äquivalent zu den oben beschriebenen einfachen Standardvariablen. So entspricht z.B. die einfache Variable A der indizierten Variablen A(1), die einfache Variable B der indizierten Variablen A(2); usw.
Die genaue Zuordnung ist aus der Tabelle auf Seite 84 ersichtlich.

Die indirekte Adressierung (Indizierung) einer Variablen gestattet es, den Namen und die Adresse einer Variablen in Abhängigkeit vom Wert einer anderen Variablen, der z.B. ein Rechenergebnis sein kann, festzulegen. Außerdem kann der Index einer Variablen auch das Ergebnis eines numerischen Ausdrucks sein.

Beispiele:

```
LET A(A) = 1243
```

Hiermit wird eine numerische Variable definiert, deren Index gleich dem ganzzahligen Anteil der im Speicher A stehenden Zahl ist. Angenommen, der Wert der Variablen A ist 7, so wird der Variablen A(7), also der Variablen G, der Wert 1234 zugewiesen.

LET A(B/5) = 789

Hierbei wird eine numerische Variable definiert, deren Index gleich dem ganzzahligen Anteil des Quotienten aus B/5 ist.

Angenommen, der Wert der Variablen B ist 134, so ist der Quotient aus 134/5 = 26,8, und somit wird der Variablen A(26) = Z ein Wert von 789 zugeordnet.

LET A$(A) ="TEXT"

Der Wert der numerischen Variablen A definiert den Index der Textvariablen A$. Dabei muß der Wert der Variablen größer als 1 sein, da die Variable A$(0) nicht existiert und die Variable A$(1) der Variablen A$ entspricht und die beiden, wie bereits erklärt, nicht gleichzeitig definiert werden können.

2.2.3.4 Besonderheiten der Variablen A

Die Variable A kann als:

a) Standardvariable A bzw. A$
b) Indizierte Standardvariable A(n) bzw. A$(n)
   (n = 1 bis 26)
c) Eindimensionale Feldvariable A(n) bzw. A$(n)
   (n = 0 bis 255)
d) Zweidimensionale Feldvariable A(n,n) bzw. A$(n,n)
   (n = 0 bis 255)

definiert werden.

Dabei gelten folgende Grundsätze:

1) Die Variable A ohne Dimensionierung

1. Diesen A-Variablen können nur numerische Werte mit zehnstelliger Mantisse, zweistelligem Exponenten und einem Vorzeichen bzw. Textwerte mit max. 7 Zeichen zugewiesen werden. Sie können nur alternativ als numerische oder Textvariablen definiert werden.

2. Der Variablenname A(0) ist unzulässig.

3. Für die indizierte Variable A(n), (n = 1 bis 26) = Standardvariablen A bis Z, wird im **PC-1401/1402** kein Speicherplatz im Hauptspeicher reserviert.

4. Für die indizierte Variable A(n), (n = 27 bis 255) = eindimensionaler Feldvariable, wird ein Speicherplatz im Hauptspeicher reserviert. Der Platz wird automatisch durch den höchsten definierten Indexwert reserviert. Wird z.B. A(n), n = 50, aufgerufen, ist der A-Vektor mit den Elementen A(27) bis A(50) automatisch vereinbart.

5. Der A-Vektor wird bezüglich seiner Elementeanzahl automatisch auf den Wert begrenzt, den er vor der Dimensionierung eines weiteren Vektors bzw. einer Matrix hatte.

Beispiel:

```
 5: A(27) = 27
10: DIM B (10)
15: DIM A (40)        oder 15: A(40) = 40
```

Nach dem Starten des Programms wird die Fehlermeldung ERROR 3 angezeigt. Der A-Vektor wurde durch die Dimensionierung des B-Vektors auf ein Element, nämlich A(27), begrenzt.
Durch den Anruf von A(40) in Zeile 15 müßte ein zweiter A-Vektor definiert werden, was aber unzulässig ist.

6. Die Variablen A(n) bzw. A$(n) (n =1 bis 26) können nicht gelöscht werden. Mit den Kommandos NEW bzw. CLEAR wird ihr Wert auf 0 (Null) bzw. auf das ASCII-Zeichen Null gesetzt.
Variablen A(n) bzw. A$(n) werden durch das Kommando RUN gelöscht.

2) Die Variable A mit Dimensionierung

1. Die dimensionierte Variable A wird behandelt wie in den Kapiteln 2.2.3.5. - 2.2.3.7. beschrieben ist.

2. Nach einer Dimensionierung der Variablen A kann auf die Standardvariablen nur noch direkt zugegriffen werden.
Nach z. B. DIM A(26) entspricht A(1) nicht mehr der Standardvariablen A und A(26) nicht mehr der Standardvariablen Z.
Das Element A(1) und die Standardvariable A haben nun verschiedene Speicherplätze.

3. Wird nach dem Aufruf einer undimensionierten Standardvariablen, z. B. A(2)=B eine Dimensionierung, z. B. DIM A(30) durchgeführt, wird ERROR 3 angezeigt.

2.2.3.5 Feldvariablen

Feldvariablen können als eindimensionale Felder (Vektoren) oder als zweidimensionale Felder (Matrizen) definiert werden, die wiederum unterschiedlich viele Elemente haben können.
Die Feldvariablen werden im Hauptspeicher abgelegt und müssen vor ihrem Aufruf wegen ihrer flexiblen Länge dimensioniert werden. Weitere Einzelheiten siehe unter dem BASIC-Befehl DIM (Abschnitt 2.5.x.x).

Der Speicherplatzbedarf für eine Feldvariable setzt sich zusammen aus:

7 BYTES für den Variablennamen
8 BYTES für ein numerisches Element
16 BYTES für ein Textelement im Standardformat

2.2.3.6 Numerische Feldvariablen

Numerische Feldvariablen können sowohl als Vektoren als auch als Matrizen definiert werden. Jedem Element der Vektoren bzw. der Matrizen kann eine vollständige Zahl, bestehend aus einer zehnstelligen Mantisse, einem zweistelligen Exponenten und einem Vorzeichen zugeordnet werden.

Vektoren werden mit dem Vektorennamen und dem Index für ein Element aufgerufen.

Der Vektorenname kann aus einem oder aus zwei Zeichen bestehen. Das erste Zeichen muß ein Buchstabe sein, das zweite Zeichen kann ein Buchstabe oder eine Ziffer sein.

Buchstabenkombinationen, die ein BASIC-Schlüsselwort oder eine Funktion ergeben, sind als Variablenname nicht erlaubt, z. B. LN, ON, OR, PI etc.

Beispiel:

```
B(0)
C1(3)
DF(80)
```

Matrizen werden mit dem Matrizennamen und den beiden Indizes für ein Element aufgerufen.

Der Matrizenname kann aus einem oder aus zwei Zeichen bestehen. Das erste Zeichen muß ein Buchstabe sein, das zweite Zeichen kann ein Buchstabe oder eine Ziffer sein.

Buchstabenkombinationen, die ein BASIC-Schlüsselwort oder eine Funktion ergeben, sind als Variablenname nicht erlaubt, z. B. LN, ON, OR, PI etc.

Beispiel:

D(2,4)
E1(1,1)
YZ(0,0)

Insgesamt stehen für Vektoren und Matrizen folgende Namen zur Verfügung:

Alle Einzelbuchstaben oder alle Zweierkombinationen aus den Buchstaben A - Z und den Ziffern 0 - 9, wobei das erste Zeichen ein Buchstabe sein muß.

Die Indizes für die Elemente liegen im Intervall 0 - 255.

Buchstabenkombinationen, die ein BASIC-Schlüsselwort oder eine Funktion ergeben, sind als Variablenname nicht erlaubt, z. B. LN, ON, OR, PI etc.

Der Index für die Vektoren- und Matrizenelemente kann als numerische Konstante oder durch eine Variable definiert werden.

Beispiele:

B(6) bzw. B(A)
H1(3,5) bzw. H1(E,B)

Der Index wird durch den jeweiligen Wert der Variablen ausgedrückt.

Vektoren und Matrizen dürfen nicht mit dem gleichen Feldnamen und nicht zweimal definiert werden. Die Namen B(n) und B(n,n) sind also nicht gleichzeitig zulässig.

Hinweis:
Bei Verwendung der Feldvariablen 'A' beachten Sie das Kapitel 2.2.3.4.

2.2.3.7 Textfeldvariablen

Textfeldvariablen können wie die numerischen Feldvariablen als Vektor oder als Matrize definiert werden. Jedem Element der Vektoren oder Matrizen können im Standardformat Texte mit maximal 16 Zeichen zugeordnet werden. Durch eine entsprechende Dimensionierung (siehe DIM-Anweisung Seite 142) kann die Größe der Elemente zwischen 1 und 80 Zeichen frei festgelegt werden.

Beispiel: DIM B$(10)*4

In diesem Beispiel können für die Elemente des B$- Vektors (B$(0) bis B$(10)) jeweils nur 4 Zeichen eingegeben werden.

Insgesamt stehen für die Vektoren und Matrizen folgende Feldnamen zur Verfügung:

Alle Einzelbuchstaben oder alle Zweierkombinationen aus den Buchstaben A - Z und aus den Ziffern 0 - 9 wobei das erste Zeichen ein Buchstabe sein muß.

Buchstabenkombinationen, die ein BASIC-Schlüsselwort oder eine Funktion ergeben, sind als Variablenname nicht erlaubt, z. B. LN, ON, OR, PI etc

Die Indizes für die Elemente liegen im Intervall 0 - 255.

Vektoren werden mit dem Vektorennamen und dem Index für ein Element aufgerufen.

Beispiele:

B$(0)
A5$(0)*20
Z$(80)*1

Matrizen werden mit dem Matrizennamen und den beiden Indizes für ein Element aufgerufen.

Beispiele:

D$(2,4)
Z1$(5,10)*12

Die Indizes für die Vektoren- und Matrizenelemente können als numerische Konstante oder als numerische Variable definiert sein, wobei der Index durch den Wert der jeweiligen Variablen bezeichnet wird.

Beispiele:

C$(A)
AE$(E,F)

Vektoren und Matrizen dürfen nicht mit dem gleichen Feldnamen und nicht zweimal definiert werden. Die Namen B$(n) und B$(n,n) sind also nicht gleichzeitig zulässig.

Hinweis:

Bei Verwendung der Texfeldvariablen 'A' beachten Sie das Kapitel 2.2.3.4.

2.2.4 ARITHMETISCHE FUNKTIONEN

Der **PC-1401/1402** bietet eine große Anzahl von arithmetischen Standard-Funktionen. Hierzu gehören:

Trigonometrische und ihre Umkehrfunktionen
Hyperbelfunktionen
Natürlicher und dekadischer Logarithmus und Umkehrfunktionen
Polarkoordinatenberechnungen

Die Ergebnisse der Funktionen sind numerische Werte. Als Parameter ist im allgemeinen ein numerischer Ausdruck zugelassen.

Fast alle Funktionsnamen können über die Funktionstasten des Rechnerteils eingegeben werden:

Funktionsname		Funktion
ABS		Absolutwert
ACS	[SHIFT] [cos⁻¹]	Arcuscosinus
AHC	[SHIFT] [hyp] [cos⁻¹]	Areacosinus
AHS	[SHIFT] [hyp] [sin⁻¹]	Areasinus
AHT	[SHIFT] [hyp] [tan⁻¹]	Areatangens
ASN	[SHIFT] [sin⁻¹]	Arcussinus
ATN	[SHIFT] [tan⁻¹]	Arcustangens
COS	[cos]	Cosinus
CUR	[SHIFT] [∛‾]	Kubikwurzel
DEG	[DEG]	Sexagesimalumrechnung
DEGREE		Winkeleinheit Grad
DMS	[SHIFT] [°'"]	Sexagesimalumrechnung
EXP	[SHIFT] [e^x]	Exponentialumrechnung
FAC	[SHIFT] [$n!$]	Fakultät
GRAD		Winkeleinheit Neugrad
HCN	[hyp] [cos]	Hyperbelcosinus
HEX/DEC		Sedezimalumrechnung
HSN	[hyp] [sin]	Hyperbelsinus
HTN	[hyp] [tan]	Hyperbeltangens
INT		Ganzzahlfunktion
LN	[ln]	Natürlicher Logarithmus
LOG	[log]	Dekadischer Logarithmus
MEM		Freier Speicherplatz
PI	[SHIFT] [π]	Kreiskonstante PI
POL	[SHIFT] [→r θ]	Polarkoordinatenumrechnung
RADIAN		Winkeleinheit
RANDOM		Zufallsgeneratoranfangswert
RCP	[1/x]	Reziprokwert
REC	[SHIFT] [→xy]	Polarkoordinatenumrechnung
RND		Zufallszahl
ROT	[SHIFT] [$\sqrt[x]{y}$]	Wurzelberechnungen (x-te Wurzel)
SGN		Vorzeichen
SIN	[sin]	Sinus
SQU	[x^2]	Quadratzahl
SQR	[√‾]	Quadratwurzel

SQR	[√‾]	Wurzelfunktion
TAN	[tan]	Tangens
TEN	[SHIFT] [10ˣ]	Exponentialfunktion 10^x
^	[SHIFT] [^]	Potenzfunktion

2.2.5 TEXTFUNKTIONEN

Mit dem **PC-1401/1402** können Sie nicht nur rein numerische Aufgaben lösen, sondern auch Texte verarbeiten. Die im folgenden Kapitel beschriebenen Funktionen sollen Ihnen diese Arbeit erleichtern. So können Sie beispielsweise Zeichenfolgen aus Texten heraustrennen und zu neuen zusammensetzen.

Die Ergebnisse der Textfunktionen sind entweder Zeichenketten, die Textvariablen zugewiesen oder in Textausdrücken weiterverarbeitet werden können, oder numerische Werte, die sich in numerischen Ausdrücken verarbeiten oder numerischen Variablen zuordnen lassen.

Folgende Textfunktionen stehen Ihnen mit dem **PC-1401/1402** zur Verfügung:

ASC Umwandlung eines Zeichens in den zugehörigen ASCII-Kode.

CHR$ Umwandlung des ASCII-Kodes in das zugehörige Zeichen.

LEFT$ Entnimmt einer Zeichenfolge eine spezifizierte Zeichenanzahl von links.

RIGHT$ Entnimmt einer Zeichenfolge eine spezifizierte Zeichenanzahl von rechts.

MID$ Entnimmt einer Zeichenfolge eine spezifizierte Zeichenanzahl aus der Mitte.

STR$ Wandelt den Wert eines numerischen Ausdrucks in eine Zeichenfolge um.

VAL Wandelt eine als Zeichenfolge eingegebene Zahl in ihren numerischen Wert um.

LEN Berechnet die Anzahl der Zeichen eines Textausdruckes.

2.2.6 PROGRAMMIEREN IN BASIC

Der **PC-1401/1402** verwendet die weit verbreitete Programmiersprache BASIC (Beginner's All-Purpose Symbolic Instruction Code). Diese Programmiersprache wurde 1960 in Dartmouth College entwickelt und hat sich bis heute insbesondere bei den Mikrocomputern durchgesetzt.

BASIC ist im Gegensatz zu den anderen Programmiersprachen einfach und leicht verständlich.

BASIC gestattet den Dialogbetrieb mit dem Rechner und ist dabei so flexibel, daß ein bestehendes Programm ohne großen Aufwand geändert werden kann.

Abgesehen von einigen Abweichungen von BASIC-Version zu BASIC-Version ist diese Programmiersprache maschinenunabhängig.

2.2.6.1 BASIC - Übersicht

Wie jede natürliche Sprache hat auch die formale Computersprache BASIC eine Grammatik und einen zugehörigen Zeichensatz.
Letzterer setzt sich aus folgenden Zeichen zusammen:

Buchstaben:	A, B, ... , Z
Zahlen:	0, 1, ... , 9
Sonderzeichen:	( ) . ; , : + - * / $ & % # @ ! ? < > = ↑

Zur Vermeidung von Verwechslungen unterscheidet man zwischen Ø (Null) und dem Buchstaben O.

BASIC kennt folgende Sprachelemente:

- numerische Konstanten
- Textkonstanten
- numerische Variablen
- Textvariablen
- dimensionierte oder Feldvariablen
- BASIC-Schlüsselwörter für
  - Standardfunktionen
  - Zuweisungen
  - Eingaben
  - Ausgaben
  - Steuerungen
  - Kommandos
- Operationszeichen (+,-,*,/,↑,>,<,=,>=,<=,<>)

Diese Sprachelemente werden zu Programmsätzen zusammengefügt, die vom Rechner der angegebenen Reihenfolge nach abgeleitet werden.

2.2.6.2 Programmerstellung

Die Programmerstellung gliedert sich im wesentlichen in drei Schritte:

PROBLEMANALYSE

UMSETZUNG IN DIE PROGRAMMIERSPRACHE
UND EINGABE IN DEN COMPUTER

TESTEN DES PROGRAMMS

Problemanalyse:

Das gestellte Problem muß zunächst analysiert und dann so aufbereitet werden, daß sich die einzelnen Teilprogramme leicht und übersichtlich programmieren lassen.

Umsetzung in die Programmiersprache und Eingabe in den Computer:

Die in die einzelnen Schritte zerlegte Aufgabenstellung wird in eine Programmiersprache (hier BASIC) übertragen und in den Computer eingegeben.

Testen des Programms:

Nur sehr selten wird das umgesetzte und eingegebene Programm auf Anhieb fehlerfrei arbeiten. Dabei können die verschiedenartigsten Fehler auftreten:

- Schreibfehler
- falsche Sprachelemente (SYNTAX-Fehler)
- falscher logischer Aufbau
- falsche Programmstruktur (fehlerhafte Problemanalyse)

Der **PC-1401/1402** bietet einige Möglichkeiten auftretende Fehler zu finden und zu beheben (siehe 2.2.6.8 Fehlermeldungen/Fehlersuche).

2.2.6.3 Programmaufbau

Bei der Übertragung der Problemanalyse in ein BASIC-Programm sind die im folgenden aufgeführten Regeln zu beachten:

- Ein vollständiges BASIC-Programm besteht aus einer Reihe von Anweisungen, die in der Reihenfolge angeordnet sein müssen, in der sie später ausgeführt werden sollen. Eine Ausnahme hierfür sind die GOTO-, GOSUB-, ON GOTO- und ON GOSUB-Anweisungen, die das Programm an eine festgelegte Stelle verzweigen können.

- Die Programmierung erfolgt zeilenweise, wobei jede Programmzeile eine oder mehrere durch Doppelpunkt getrennte Programmanweisungen enthalten kann.

- Die Anweisungen dürfen nicht länger als eine Zeile sein, d.h., sie können nicht in der nächsten Zeile fortgesetzt werden. Eine Zeile kann bis zu 79 Zeichen enthalten, wovon jeweils 16 Zeichen auf dem Display angezeigt werden. Die restlichen Zeichen einer Zeile können mit den (Cursor-)Tasten [←] oder [→] sichtbar gemacht werden.

  Hinweis: BASIC-Schlüsselwörter wie LET, SIN, COS, PRINT usw. werden nach Betätigung der [ENTER]-Taste intern abgekürzt. Bei der Eingabe einer sehr langen Programmzeile kann man vorerst nur 79 Zeichen eingeben. Nach Drücken der [ENTER]-Taste werden die Schlüsselwörter abgekürzt, und man kann zusätzlich noch einige Zeichen eingeben. Die Eingabe muß wieder mit [ENTER] abgeschlossen werden.

- Jede Zeile muß mit einer positiven ganzen Zahl (der Zeilennummer) beginnen. Eine Zeilennummer darf nur einmal im Programm verwendet werden. Die Programmzeilen werden vom Rechner in aufsteigender Reihenfolge ausgeführt, wenn nicht durch Steueranweisungen (z.B.GOTO) eine andere Reihenfolge festgelegt wird.
  Die Numerierung muß nicht lückenlos sein, es ist sogar zweckmäßig, z.B. Zehnerschritte zu wählen, da damit die Möglichkeit geboten ist, nachträglich noch Programmzeilen einzufügen.
  Die Zeilennummern dürfen beim PC-1401/PC-1402 im Bereich von 1 bis 65279 gewählt werden.

2.2.6.4 Eingabe eines Programms

Die Programmeingabe erfolgt im PRO-Mode. Ein sich eventuell noch im Speicher befindliches Programm kann mit der NEW-Anweisung gelöscht werden.

Beispiel einer Programmeingabe:

```
NEW

10 A = 15
20 B = 3
30 C = A + B
40 PRINT C
50 END
```

Jede Programmzeile muß mit [ENTER] abgeschlossen werden. Der Rechner fügt dann einen Doppelpunkt zwischen die Zeilennummer und der ersten Anweisung ein.

Beispiel:

10 A = 15 [ENTER]

Anzeige:

10:A = 15

Oberprüfen des Programms, Editieren und Auflisten

Ist die Programmeingabe abgeschlossen, so kann man im PRO-Mode den Inhalt der einzelnen Zeilen noch einmal überprüfen.

Betätigt man die [↑] -Taste, wird die vorhergehende Programmzeile angezeigt.
Betätigt man die [↓] -Taste, wird die folgende Programmzeile angezeigt.

Mit der LIST-Anweisung kann man genau spezifizierte Zeilen zur Anzeige bringen.

Beispiele:

```
LIST          Die erste Zeile eines Programms wird angezeigt.
LIST 100      Die Programmzeile 100 wird angezeigt.
```

Mit der LLIST-Anweisung kann das Programm über den Drucker (Option CE-126P) auf Papier 'gelistet' werden.

Beispiele:

```
LLIST          Das gesamte Programm wird ausgedruckt.
LLIST 100      Die Programmzeile 100 wird ausgedruckt.
LLIST 100,200  Die Programmzeilen von 100 bis 200 werden ausgedruckt.
LLIST 100,     Das Programm wird ab Programmzeile 100 ausgedruckt.
LLIST ,100     Das Programm wird bis Programmzeile 100 ausgedruckt.
```

Beispiel:

```
10 A = 15
20 B = 3
30 C = A + B
40 PRINT C
50 END
```

Durch Eingabe von LIST 30 (nur im PRO-Mode möglich) wird am Display folgendes angezeigt:

```
30:C = A + B
```

Die Zeile 40 kann jetzt durch Drücken der [↓]-Taste angezeigt werden.

```
40:PRINT C
```

2.2.6.5 Korrektur einer Zeile

Die Korrektur einer Zeile (nur im PRO-Mode) ist sehr einfach und kann auf zwei verschiedene Arten erfolgen:

1. Oberschreiben der Zeile:
   Man gibt die neue Zeile mit derselben Zeilennummer ein. Die alte Version der Programmzeile wird dadurch überschrieben.

2. Editieren:
   Man bringt die zu editierende Zeile mit LIST zur Anzeige und drückt eine der beiden Cursor-Tasten ([←] oder [→]). Damit wird der Doppelpunkt zwischen der Zeilennummer und der ersten Programmanweisung unterdrückt und die Zeile zur Korrektur freigegeben. Jetzt können Sie mit Hilfe der [SHIFT][DEL]-oder der [INS]-Taste Zeichen in der Zeile löschen oder einfügen. Auch ein Oberschreiben einzelner Zeichen ist möglich.

Beispiel:

In dem oben angeführten Beispiel soll die Zahl 15 durch die Zahl 17 ersetzt werden. Hierzu geben Sie ein:

LIST 10 [ENTER] (Zeile 10 anzeigen)

[→] (Beginn Änderung)

[→] [→] [→] (Cursor auf die '5')

7 [ENTER] ('5' durch '7' ersetzen)

2.2.6.6 Löschen einer Zeile

Das Löschen einer Programmzeile erfolgt im PRO-Mode. Eine Zeile wird ersatzlos gelöscht, wenn man nur die Zeilennummer eingibt und die [ENTER]-Taste drückt.

Beispiel:

```
10 A = 15
20 B = 3
30 C = A + B
40 PRINT C
50 END
```

Eingabe:

10 [ENTER]
20 [ENTER]

Listet man nun das Programm, so sieht man, daß die Zeilen 10 und 20 gelöscht wurden.

2.2.6.7 Programmausführung

Die Programmausführung kann nur im RUN-Mode erfolgen. Der Programmstart kann durch drei verschiedene Anweisungen erfolgen (siehe RUN/GOTO/Definable Keys).

Nach Eingabe des Startkommandos beginnt der **PC-1401/1402** mit der Abarbeitung der einzelnen Programmzeilen. Während der Programmausführung ist auf der Anzeige das Wort "BUSY" sichtbar. Dies erlischt entweder am Programmende oder, wenn eine Anzeige am Display erfolgt. Solange 'BUSY' aufleuchtet, nimmt der Computer keine Eingaben von der Tastatur an. Nach Beendigung des Programms wird am Display das 'Bereitschaftssymbol' (>) angezeigt.

Zur Unterbrechung einer Programmausführung dient die [BRK] -Taste. Am Display wird folgendes angezeigt:

BREAK IN XX wobei XX die Zeilennummer ist, die vor der Unterbrechung bearbeitet wurde.

Will man nach einer Unterbrechung das Programm fortsetzen, so genügt es, die CONT(inue)-Anweisung einzugeben:

CONT

Die Programmausführung wird fortgesetzt.

2.2.6.8 Fehlermeldung/Fehlersuche

Tritt während der Programmausführung ein Fehler auf, so wird dieser vom **PC-1401/1402** erkannt und gemeldet. Die Fehlermeldung wird am Display in folgender Form angezeigt:

ERROR Fehlerkode IN Zeilennummer

Die Liste der Fehlermeldungen und ihre Erklärung finden Sie numerisch geordnet im Anhang.

Gleichzeitig mit dem Fehlerkode wird die Nummer der Zeile, in der der Fehler auftrat, angezeigt. Diese Zeile kann nun auch im RUN-Mode am Display zur Anzeige gebracht werden. Löschen Sie die Fehlermeldung mit der [C·CE] -Taste und drücken Sie die Taste [↑] . Die Zeile erscheint auf der Anzeige, und der Cursor blinkt auf dem fehlerhaften Sprachelement. Will man die Zeile korrigieren, muß man in den PRO-Mode umschalten und wie in Abschnitt 2.2.6.5 beschrieben vorgehen.

Beispiel:

```
10 A = 2
20 B = 10
30 C = B/A
40 PRINT C,
50 END
```

Bei der Ausführung diese Programm erscheint:

ERROR 1 IN 40

Die PRINT-Liste in Zeile 40 ist unvollständig. Schreiben Sie hinter das Komma ein A.

```
40 PRINT C,A
```

Das Programm ist jetzt fehlerfrei.

2.2.7 PC-1401/1402-BASIC – BEFEHLSVORRAT

Im folgenden Kapitel wird der BASIC-Sprachumfang des **PC-1401/1402** ausführlich beschrieben. Dabei sind für jeden Befehl kurz die Funktion, die Syntax, eventuelle Bemerkungen und ein Beispiel angegeben.

Hinweis zur Syntax:

Die Bedeutung des jeweiligen Sprachelements wird in spitzen Klammern angegeben.

2.2.7.1 Fest vorprogrammierte Schlüsselwörter und arithmetische Funktionen

Die am häufigsten verwendeten BASIC-Anweisungen können durch die Tastenkombination [SHIFT] und einen Buchstaben der beiden unteren Tastenreihen abgerufen werden.

Außerdem lassen sich fast alle arithmetischen Funktionen direkt über die Funktionstasten auf der rechten Seite des Rechners direkt ins BASIC übernehmen.

Die Zuordnung der Anweisungen ist aus der folgenden Tabelle zu entnehmen:

Tastenbetätigung		BASIC-Anweisung
[SHIFT]	A	INPUT
[SHIFT]	S	IF
[SHIFT]	D	THEN
[SHIFT]	F	GOTO
[SHIFT]	G	FOR
[SHIFT]	H	TO
[SHIFT]	J	STEP
[SHIFT]	K	NEXT
[SHIFT]	L	LIST
[SHIFT]	,	RUN
[SHIFT]	Z	PRINT
[SHIFT]	X	USING
[SHIFT]	C	GOSUB
[SHIFT]	V	RETURN
[SHIFT]	B	DIM
[SHIFT]	N	END
[SHIFT]	M	CSAVE
[SHIFT]	SPC	CLOAD

Hinweise:

1. Da diese Begriffe im Rechner kodiert als Schlüsselwörter für BASIC-Anweisungen gespeichert werden, ist es z.B. nicht möglich,

   aus [L] [SHIFT] [L]

   ein LLIST zusammenzusetzen.

2. Eine weitere Möglichkeit zur Reduzierung der Eingabearbeit ist durch die Verwendung von Abkürzungen gegeben. Eine Liste der Abkürzungen finden Sie im Anhang.

Die Zuordnung der arithmetischen Befehle ist aus der in Abschnitt 2.2.4 dargestellten Tabelle zu entnehmen.

ABS

Funktion:

Ermittelt den Absolutbetrag eines numerischen Ausdrucks.

Syntax:

ABS numerischer Ausdruck

Bemerkungen:

Der Absolutwert ist der Wert eines numerischen Ausdrucks ohne Berücksichtigung seines Vorzeichens.

Beispiel:

```
10 A = ABS (-6)
20 B = ABS (6)
30 C = ABS (-3 * 7)
40 PRINT A;"   ";B;"   ";C
```

Ausgabe nach RUN:

```
6.  6.  21.
```

ACS

Funktion:

Berechnet den Arcuscosinus eines numerischen Ausdrucks in der angegebenen Winkeleinheit.

Syntax:

ACS numerischer Ausdruck

Bemerkungen:

Die Berechnung der Arcuscosinusfunktion kann in folgenden drei Winkeleinheiten erfolgen:
DEGREE (0 bis 90°)
GRAD (0 bis 100^g)
RADIAN (0 bis PI/2)
Die gewählte Winkeleinheit wird am Display angezeigt.

Beispiel:

```
10 DEGREE : INPUT X
20 LET Y = ACS X
30 PRINT"X=";X;" ";"ACS X=";Y
```

Ausgabe nach RUN:

```
?                    1   ENTER

X=1. ACS X=0.
```

AHC

Funktion:

Berechnet den Areacosinus eines numerischen Wertes in der angegebenen Winkeleinheit.

Syntax:

AHC numerischer Ausdruck

Beispiel:

```
10 INPUT X
20 LET Y = AHC X
30 PRINT"X=";X;" ";"AHC X=";Y
```

Ausgabe nach RUN:

```
?                              1   ENTER

X=1. AHC X=0.
```

AHS

Funktion:

Berechnet den Areasinus eines numerischen Wertes in der angegebenen Winkeleinheit.

Syntax:

AHS numerischer Ausdruck

Beispiel:

```
10 INPUT X
20 LET Y = AHS X
30 PRINT"X=";X;" ";"AHS X=";Y
```

Ausgabe nach RUN:

```
?                              1   ENTER

X=1. AHS X=0.881
```

AHT

Funktion:

Berechnet den Areatangens eines numerischen Wertes in der angegebenen Winkeleinheit.

Syntax:

AHT numerischer Ausdruck

Beispiel:

```
10 INPUT X
20 LET Y = AHT X
30 PRINT Y
```

Ausgabe nach RUN:

```
?                    0.5   ENTER

 5.493061443E-01
```

AND

Funktion:

"UND"-Verknüpfung zweier numerischer Ausdrücke.

Syntax:

numerischer Ausdruck 1 **AND** numerischer Ausdruck 2

Bemerkungen:

Das Ergebnis einer "UND"-Verknüpfung ist wahr ("1"), wenn der Ausdruck 1 und der Ausdruck 2 wahr sind, sonst ist es falsch ("0").

Wahrheitstabelle:

A	B	A AND B
0	0	0
1	0	0
0	1	0
1	1	1

Bei der "UND"-Verknüpfung werden die Ausdrücke als 16stellige Binärzahlen interpretiert. Die Ausdrücke müssen dabei im Bereich zwischen -32768 und +32767 liegen. Liegt einer der Ausdrücke nicht in diesem Bereich, erfolgt die Fehlermeldung ERROR 3. Das Ergebnis einer "UND"-Verknüpfung weist in jeder binären Stelle eine "1" auf, in der im Ausdruck 1 und im Ausdruck 2 eine "1" steht, sonst eine "0".

Beispiel:

X AND 1

Ergebnis	16-Bit-Darstellung	
0 wenn X = 0	00000000 00000000	(X)
	00000000 00000001	(1)
	00000000 00000000	(X AND 1)
1 wenn X = 1	00000000 00000001	(X)
	00000000 00000001	(1)
	00000000 00000001	(X AND 1)

34 AND 70

Ergebnis	16-Bit-Darstellung	
2	00000000 00100010	(34)
	00000000 01000110	(70)
	00000000 00000010	(34 AND 70 = 2)

AREAD

Funktion:

Liest den vor dem Programmstart angezeigten numerischen oder Textausdruck in eine Variable ein.

Syntax:

AREAD numerische Variable
AREAD Textvariable

Bemerkungen:

AREAD muß die **erste** Anweisung im Programm sein. Der vor dem Programmstart angezeigte numerische oder Textausdruck wird der auf die AREAD-Anweisung folgenden Variablen zugewiesen. Das Programm muß über eine der Definable Keys (Definierbare Tasten) gestartet werden.

Hinweise:

1. Wenn nur das Bereitschaftszeichen am Anfang der Programmausführung angezeigt wird, so wird der auf die AREAD-Anweisung folgenden Variablen der Wert 0 (Null) bzw. ASCII-Null zugewiesen.
2. Wenn vor dem Programmstart eine PRINT-Liste aus einem vorher durchlaufenen Programm angezeigt wird, so wird nur der letzte Ausdruck der Liste gespeichert. Dabei ist zu beachten, daß die Variable nach AREAD der Art des Ausdrucks entspricht.

Beispiel:

```
10 "S" AREAD X
20 PRINT "SIN(";X;")=";SINX
```

Geben Sie ein:

```
1 . 5
DEF S
```

Ausgabe in Zeile 20:

```
SIN(1.5)=2.61769
```

ASC

Funktion:

Ergibt den ASCII-Kode eines Zeichens bzw. des ersten Zeichens eines Textausdrucks.

Syntax:

ASC "Zeichen"
ASC Textausdruck

Bemerkungen:

Die ASC-Funktion wandelt das erste Zeichen eines Textausdrucks in die dezimale Darstellung des ASCII-Codes um.
Das Ergebnis der ASC-Funktion ist ein numerischer Wert im Bereich zwischen 32 und 96 (siehe ASCII-Code-Tabelle im Anhang).

Beispiel:

(1)
```
10 A=ASC"A"
20 PRINT A
```

Ausgabe nach RUN:

```
65.  = ASCII-Code des Zeichens A.
```

(2)
```
10 A$="SHARP"
20 A=ASC A$
30 PRINT A
```

Ausgabe nach RUN:

```
83.  = ASCII-Code des Zeichens S.
```

ASN

Funktion:

Berechnet den Arcussinus eines numerischen Ausdrucks in der angegebenen Winkeleinheit.

Syntax:

ASN numerischer Ausdruck

Bemerkungen:

Die Berechnung der Arcussinusfunktion kann in folgenden drei Winkeleinheiten erfolgen:
DEGREE (0 bis 90°)
GRAD (0 bis 100^g)
RADIAN (0 bis PI/2)
Die gewählte Winkeleinheit wird am Display angezeigt.

Beispiel:

```
10 DEGREE : INPUT X
20 LET Y = ASN X
30 PRINT"X=";X;" ";"ASN X=";Y
```

Ausgabe nach RUN:

```
?                    1   ENTER

X=1. ASN X=90.
```

ATN

Funktion:

Berechnet den Arcustangens eines numerischen Ausdrucks in der angegebenen Winkeleinheit.

Syntax:

ATN numerischer Ausdruck

Bemerkungen:

Die Berechnung der Arcustangensfunktion kann in folgenden drei Winkeleinheiten erfolgen:
DEGREE (0 bis 90°)
GRAD (0 bis 100^{g})
RADIAN (0 bis PI/2)
Die gewählte Winkeleinheit wird am Display angezeigt.

Beispiel:

```
10 DEGREE : INPUT X
20 LET Y = ATN X
30 PRINT"X=";X;" ";"ATN X=";Y
```

Ausgabe nach RUN:

```
?                    1   ENTER

X=1. ATN X=45.
```

BEEP

Funktion:

Steuert den akustischen Signalgeber (Pips).

Syntax:

BEEP numerischer Ausdruck < Anzahl der Signaltöne >

Bemerkungen:

Das Signal besteht aus einer von kurzen Pausen unterbrochenen Folge von Tönen gleicher Frequenz.

Beispiel:

```
10 A=10
20 B=20
30 C=SQR (A+B)
40 BEEP 5
50 PRINT C
60 END
```

Der Ausgabe des Ergebnisses in Zeile 50 werden fünf Signaltöne vorangestellt.

CHR$

Funktion:

Ermittelt den ASCII-Code eines Zeichens.

Syntax:

CHR$ (numerischer Ausdruck)

Bemerkungen:

Das Ergebnis der CHR$-Funktion ist ein Textzeichen. Der Wert des numerischen Ausdrucks im Bereich von 32 < Ausdruck < 96 definiert den ASCII-Code eines Zeichens (siehe ASCII-Code-Tabelle im Anhang).

Beispiel:

Aufstellung einer Code-Tabelle:

```
10 FOR I = 33 TO 95
20 PAUSE I;"--=--";CHR$ I
30 NEXT I
```

Ausgabe nach RUN:

```
33.--=--!
34.--=--"
.
.
.
95.--=--_
```

CLEAR

Funktion:

Löscht alle Variablen und Felder aus dem Hauptspeicher und setzt die Standardvariablen auf 0 (Null).

Syntax:

CLEAR

Bemerkungen:

Das Programm bleibt im Gegensatz zum NEW-Befehl erhalten.

Beispiel:

```
10 LET X=17
20 PAUSE "X=";X
30 CLEAR
40 PRINT "X=";X
50 END
```

Ausgabe nach RUN:

```
X=17.
X=0.
```

CLOAD (nur mit Option CE-126P)

Funktion:

Laden der Programme von Band.

Syntax:

CLOAD
CLOAD Textausdruck

Bemerkungen:

Mit der CLOAD-Anweisung werden auf Band gespeicherte Programme gesucht und geladen. Die CLOAD-Anweisung ist nur als direktes Kommando möglich. Befindet sich ein Programm im Hauptspeicher, so ist dieses vor einer CLOAD-Anweisung zu sichern, da die Anweisung den gesamten Hauptspeicher löscht, bevor ein neues Programm geladen wird.

Der der CLOAD-Anweisung folgende Textausdruck bezeichnet den Programmnamen. Wird dieser beim Ladevorgang gefunden, so wird das Programm in den Hauptspeicher geladen.

Erfolgt die Eingabe von CLOAD ohne einen Textausdruck (Programmnamen), so wird das nächste auf Band gespeicherte Programm geladen.

Befindet sich das Programm nicht auf der im Recorder befindlichen Cassette, so sucht der Computer auch dann noch nach dem Programmnamen, wenn das Band schon abgelaufen ist. In diesem Fall muß der Ladevorgang mit der BRK -Taste abgebrochen werden.

Tritt während des Ladevorgangs ein Fehler auf, so ist das im Computer befindliche Programm nicht verwendbar. Der Ladevorgang muß wiederholt werden.

Beispiele:

```
CLOAD

CLOAD"PROG.1"
```

CLOAD? (nur mit Option CE-126P)

Funktion:

Vergleich des im Hauptspeicher gespeicherten Programms mit dem auf Band gespeicherten Programm.

Syntax:

CLOAD?
CLOAD? Textausdruck

Bemerkungen:

Mit der CLOAD?-Anweisung wird das im Hauptspeicher gespeicherte Programm mit dem auf Band gespeicherten Programm verglichen. Tritt dabei ein Fehler auf, so wird am Display die Fehlermeldung ERROR 8 angezeigt.

Der Ablauf der CLOAD?-Anweisung ist identisch mit dem der CLOAD-Anweisung.

Ein eventuell verwendetes PASS-Wort wird mit CLOAD? nicht geprüft.

Beispiele:

```
CLOAD?

CLOAD? "PROG.1"
```

CONT

Funktion:

Die Programmausführung wird nach einer Unterbrechung durch STOP oder durch Drücken der BREAK-Taste fortgesetzt.

Syntax:

CONT

Bemerkungen:

Alle Zustände des Rechners (FOR...NEXT, GOSUB, DIM usw.) bleiben erhalten. Dadurch ist es möglich, einen Programmablauf zu unterbrechen (STOP oder BREAK), um die aktuellen Werte von Variablen zu überprüfen und gegebenenfalls zu verändern. Die Programmfortsetzung erfolgt mit der CONT-Anweisung.

Beispiel:

```
 5 A=0
10 FOR I=1 TO 20
20 A=A+1
30 PAUSE A
40 NEXT I
50 END
```

Ausgabe nach RUN:

```
1.
2.
3.

BREAK-Taste drücken

BREAK IN XX          XX...Zeilennummer, in der der Programmablauf
                          unterbrochen wurde.

CONT
4.
5.
6.
.
.
.
20.
```

COS

Funktion:

Berechnet den Cosinus eines numerischen Ausdrucks in der angegebenen Winkeleinheit.

Syntax:

COS numerischer Ausdruck

Bemerkungen:

Die Berechnung der Cosinusfunktion kann in folgenden drei Winkeleinheiten erfolgen:
DEGREE (0 bis 90°)
GRAD (0 bis 100ᵍ)
RADIAN (0 bis PI/2)
Die gewählte Winkeleinheit wird am Display angezeigt.

Beispiel:

```
10 DEGREE : INPUT X
20 LET Y = COS X
30 PRINT Y
```

Ausgabe nach RUN:

? 1 ENTER

9.998476952E-01

CSAVE (nur mit Option CE-126P)

Funktion:

Speichern eines Programms auf Band.

Syntax:

```
CSAVE
CSAVE "Programmname"
CSAVE ,"PASS-Wort"
CSAVE "Programmname","PASS-Wort"
```

Bemerkungen:

Mit der CSAVE-Anweisung können Programme auf Band gespeichert werden. Bei der CSAVE-Anweisung gibt es folgende vier Eingabemöglichkeiten:

CSAVE Alle im Computer gespeicherten Programme werden auf Band gespeichert.

CSAVE "Programmname" Alle im Computer gespeicherten Programme werden unter dem angegebenen Programmnamen auf Band gespeichert.

CSAVE ,"PASS-Wort" Alle im Computer gespeicherten Programme werden mit dem PASS-Wort auf Band gespeichert. Programme, die mit einem PASS-Wort auf Band gespeichert wurden und wieder geladen werden, können nur nach Eingabe des zugehörigen PASS-Wortes wieder gelistet oder bearbeitet werden.

CSAVE "Programmname","PASS-Wort" Alle im Computer gespeicherten Programme werden unter dem angegebenen Programmnamen mit dem PASS-Wort auf Band gespeichert.

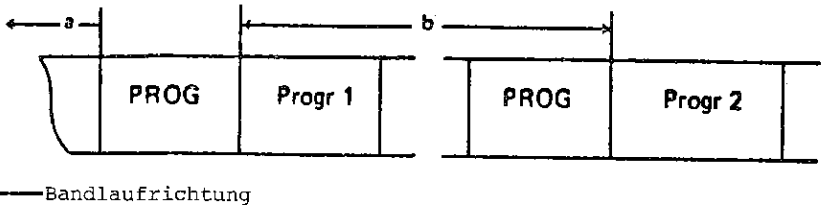
Die CSAVE-Anweisung kann im PRO- und im RUN-Mode eingegeben werden. Je nach Programmlänge dauert der Speichervorgang einige Minuten.

Im Anschluß an das Speichern des Programms sollte man mit der CLOAD?-Anweisung überprüfen, ob nicht, bedingt durch einen Fehler beim Speichervorgang oder durch einen Bandfehler, Informationen verlorengegangen sind.

Hinweise:

1. Wird ein neues Programm so auf dem Band gespeichert, daß es ein anderes Programm auch nur teilweise überlappt, wird die ursprüngliche Information gelöscht. Dies führt zu einem Fehler beim Laden des ursprünglichen Programms.

2. Haben zwei Programme denselben Programmnamen, z.B. "PROG", so lädt der Computer das zuerst aufgefundene Programm in den Speicher. Befindet sich der Tonkopf in bezug auf das Band in der Abbildung im Bereich a, so wird das erste Programm mit dem Namen "PROG" geladen. Befindet er sich jedoch im Bereich b, so wird das zweite Programm geladen.



3. Zweckmäßigerweise sollte für jede Cassette der Programmbibliothek ein Karteiblatt angelegt werden, aus dem die Programmnamen, Hinweise auf Informationstyp (Programm, Daten), Zählerstand und kurze Programmbeschreibung hervorgehen. Damit läßt es sich vermeiden, daß Programme oder Daten nur mühselig und mit sehr großem Aufwand wiedergefunden werden können.

Beispiele:

```
CSAVE

CSAVE "PROG 1"

CSAVE ,"GEHEIM"

CSAVE "PROG 1","GEHEIM"
```

CUR

Funktion:

Berechnet die Kubikwurzel eines numerischen Ausdrucks.

Syntax:

CUR numerischer Ausdruck

Bemerkungen:

-

Beispiel:

```
10 X=27
20 Y=CUR X
30 PRINT"X=";X;" CUR X=";Y
40 END
```

Ausgabe nach RUN:

```
X=27. CUR X=3.
```

DATA

Funktion:

Definiert Datenfelder für die READ-Anweisung.

Syntax:

DATA numerischer Ausdruck
DATA Textausdruck
DATA Liste von numerischer Ausdrücke
DATA Liste von Textausdrücken
DATA Liste von numerischen Ausdrücken und Textausdrücken

Bemerkungen:

Die DATA-Anweisung gewinnt ihre Bedeutung erst im Zusammenhang mit der READ-Anweisung.
Mit der DATA-Anweisung werden Daten innerhalb eines Programms gespeichert und zum Abruf mit der READ-Anweisung bereitgestellt. DATA-Anweisungen können in beliebiger Anzahl und an beliebiger Stelle im Programm auftreten und werden während des Programmablaufs übersprungen. Die DATA-Anweisung kann eine beliebige Anzahl von Konstanten enthalten (bis zur maximalen Zeilenlänge), die jeweils durch ein Komma getrennt werden. DATA-Anweisungen können wiederholt von Anfang an gelesen werden, wenn man die RESTORE-Anweisung benutzt.

Hinweis:
Wichtig ist es, daß die Daten-Typen in einer DATA-Anweisung mit den entsprechenden Variablen-Typen in der READ-Anweisung übereinstimmen.

Beispiele:

```
10 DATA 12,13,14
20 DATA "TEXTE","KOELN"
30 DATA 1,"BEI","SPIEL"
40 DATA A + B, X/Y
```

DEGREE/GRAD/RADIAN

Funktion:

Festlegung der Winkeleinheit.

Syntax:

DEGREE
GRAD
RADIAN

Bemerkungen:

Die Winkel werden wie folgt angegeben:

Winkeleinheit	Anzeige	Viertelkreis	Einheit
DEGREE (Grad)	DEG	(0 bis 90°)	Grad
GRAD (Neugrad)	GRAD	(0 bis 100^g)	Gon
RADIAN (Radiant)	RAD	(0 bis PI/2)	rad

Die gewählte Winkeleinheit wird im Display angezeigt.
Die Winkeleinheit kann auch programmgesteuert verändert werden.

Beispiel:

```
10 WAIT 200
20 DEGREE : PRINT SIN 45
30 GRAD : PRINT SIN 50
40 RADIAN : PRINT SIN (PI/4)
```

Ausgabe nach RUN:

```
 7.071067812E-01
 7.071067812E-01
 7.071067812E-01
```

DEG

Funktion:

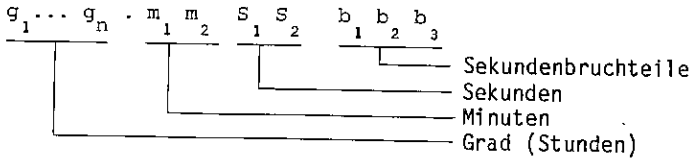
Umrechnung vom Sexagesimalsystem ins Dezimalsystem.

Syntax:

DEG numerische Konstante
DEG numerische Variable

Bemerkungen:

Sexagesimal geteilte Winkel oder Zeiten werden in folgendem Format verarbeitet:



Beispiel:

Umrechnung von 15°24'45'' in Dezimalgrad

```
10 A=DEG 15.2445
20 PRINT A
30 END
```

Ausgabe nach RUN:

```
        15.4125
```

DIM

Funktion:

Dimensionierung von ein- bzw. zweidimensionalen Feldern (Arrays).

Syntax:

DIM Feldname (num. Ausdruck)
DIM Feldname (num. Ausdruck, num. Ausdruck)

DIM Text-Feldname (num. Ausdruck)
DIM Text-Feldname (num. Ausdruck) * num. Ausdruck
DIM Text-Feldname (num. Ausdruck, num. Ausdruck)
DIM Text-Feldname (num. Ausdruck, num. Ausdruck) * num. Ausdruck

Bemerkungen:

Die DIM-Anweisung legt für einen Vektor (eindimensional) oder für eine Matrix (zweidimensional) die Gesamtlänge, die Dimension und die Indexbereiche fest.

Die DIM-Anweisung reserviert für das Feld (Text-Feld) den notwendigen Speicherplatz im Hauptspeicher.

Der Feldname (Text-Feldname) bezeichnet das gesamte Feld.

Die Werte der numerischen Ausdrücke spezifizieren die Anzahl der Zeilen eines eindimensionalen Feldes bzw. die Anzahl der Zeilen und Spalten eines zweidimensionalen Feldes.

Bei Textfeldern kann mit einem weiteren numerischen Ausdruck (nach dem Asterix-Zeichen '*') die Textfeldlänge (Zeichenanzahl) festgelegt werden.

Der Zugriff auf ein Feldelement während des Programmlaufs erfolgt durch die Angabe des Feldnamens (Text-Feldnamens) und ein oder zwei Indizes (numerische Ausdrücke) je nach DIM-Vereinbarung.

Im Unterschied zu einer normalen Variablen sieht die LET-Anweisung beispielsweise wie folgt aus:

```
LET AX (5) = 7
LET AR (1,1) = 3.24
```

Die Dimensionierung ist erstens durch den zur Verfügung stehenden Speicherplatz und zweitens durch die Indizes, die im Bereich zwischen 0 und 255 liegen müssen, begrenzt.

Bei Textfeldern darf die Textfeldlänge eines Elements 80 Zeichen nicht überschreiten. Wird die Textfeldlänge bei der Dimensionierung nicht festgelegt, so wird dem Textfeld automatisch eine Länge von 16 Zeichen zugeordnet.

Bei der Dimensionierung wird den numerischen Feldern der Wert Null und den Textfeldern der ASCII-Wert Null zugeordnet.
Ein Feld darf innerhalb eines Programms nur einmal definiert werden, ansonsten wird die Fehlermeldung ERROR 5 angezeigt.

Da Null (0) ein legaler Wert für einen Index ist, hat das Feld jeweils eine um 1 größere Anzahl von Zeilen bzw. Spalten als die Werte der DIM-Anweisung.

Numerische und Textfelder dürfen den gleichen Namen haben; die DIM-Anweisung:

```
DIM B(3,3),B$(2,5)
```

ist zulässig.
Für die Dimensionierung von Vektoren und Matrizen können Feldnamen, bestehend aus einem oder zwei Zeichen verwendet werden. Dabei muß das erste Zeichen ein Buchstabe sein, das zweite Zeichen kann aus einem Buchstaben oder einer Ziffer bestehen.

Buchstabenkombinationen, die ein BASIC-Schlüsselwort oder eine Funktion ergeben, sind als Variablenname nicht erlaubt, z. B. LN, ON, OR, PI etc.

Die Indizes liegen im intervall 0 - 255

Hinweise:

1. Die Variable A nimmt eine Sonderstellung ein.
   Einzelheiten siehe Abschnitt 2.2.3.4.

2. Es empfiehlt sich, der Übersicht halber alle Felder möglichst vor der ersten Anweisung des Hauptprogramms zu dimensionieren.

3. Nach Fertigstellung des Programms sollten die DIM-Anweisungen anhand des noch verfügbaren Speicherplatzes überprüft werden. Dabei gilt:

   Anzahl der dimensionierten Elemente INT ((MEM -7)/8)
   Handelt es sich um eine Textfeldvariable, muß anstelle der 8 eine 16 bzw. die gewünschte Textfeldvariable eingegeben werden.

4. Mit dem PC-1401/1402 ist es möglich, eine indizierte Variable als Indizes einer zweidimensionalen Feldvariablen zu verwenden.

```
B(A*B,C(0))=10
B(C(0),5)=10
B(4,A(30))=10
```

Beispiele:

(1) 10 DIM X(4)

Die Programmzeile 10 vereinbart ein numerisches eindimensionales Feld mit fünf Elementen, das wie folgt aussieht:

X (0)	X (1)	X (2)	X (3)	X (4)

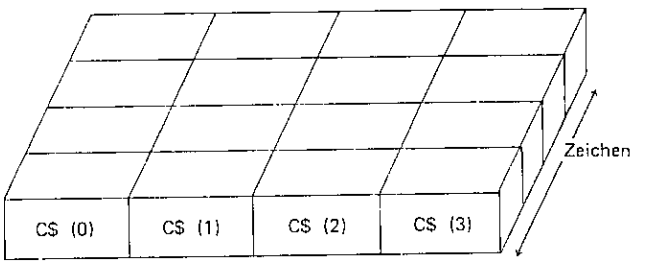
(2) 20 DIM B(2,3)

Die Programmzeile 20 vereinbart ein zweidimensionales Feld mit 12 Elementen.

AB (0, 0)	AB (0, 1)	AB (0, 2)	AB (0, 3)
AB (1, 0)	AB (1, 1)	AB (1, 2)	AB (1, 3)
AB (2, 0)	AB (2, 1)	AB (2, 2)	AB (2, 3)

(3) 30 DIM C$(3) * 4

Die Programmzeile 30 vereinbart ein Textfeld mit vier Elementen mit einer Textfeldlänge von je vier Zeichen.



(4)
```
10 DIM V(3), P(2)
20 I = 4 : J = 5
30 DIM W$ (I * J)
```

In Programmzeile 30 wird ein Textvektor dimensioniert, dessen Index sich aus dem Produkt der Variablen I und J errechnet (W$(20)).

(5)
```
10 DIM S(9,1)
20 FOR I = 0 TO 90 STEP 10
30 S(I/10,1) = SIN I
40 S(I/10,0) = I
50 NEXT I
60 FOR J = 0 TO 9
70 PRINT S(J,0),S(J,1)
80 NEXT J
```

In Zeile 10 wird folgendes Feld vereinbart:

S(0, 0)	S(0, 1)
S(9, 0)	S(9, 1)

In der Programmschleife von Zeile 20 bis Zeile 50 nimmt die Schleifenvariable I die Werte 0, 10, 20 bis 90 an. In den Zeilen 30 und 40 werden die Indizes des Feldes S bestimmt.
Nach Durchlaufen der Schleife enthält S den Winkel und den Sinus des Winkels.

Ausgabe nach RUN:

```
 0.       0.   ENTER
10.  1.7E-01   ENTER
20.  3.4E-01   ENTER
30.       0.5  ENTER
40.  6.4E-01   ENTER
50.  7.6E-01   ENTER
60.  8.6E-01   ENTER
70.  9.3E-01   ENTER
80.  9.8E-01   ENTER
90.       1.
```

DMS

Funktion:

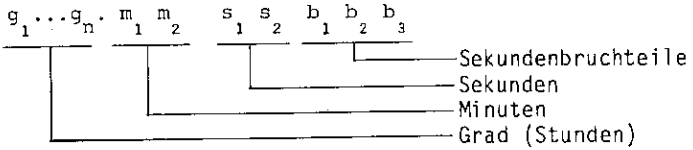
Umrechnung eines numerischen Ausdrucks vom Dezimalsystem ins Sexagesimalsystem.

Syntax:

DMS numerischer Ausdruck

Bemerkungen:

Sexagesimal geteilte Winkel oder Zeiten werden in folgendem Format verarbeitet:



Beispiel:

Umrechnung von 15.4125 (Grad) in Grad/Minuten/Sekunden

```
10 A=DMS 15.4125
20 PRINT A
30 END
```

Ausgabe nach RUN:

```
         15.2445
```

END

Funktion:

Beendet die Programmausführung.

Syntax:

END

Bemerkungen:

Die END-Anweisung steht normalerweise am Ende eines Programmes und beendet die Ausführung.

Unter Umständen kann die END-Anweisung auch weggelassen werden, da die Programmausführung automatisch nach Abarbeiten der letzten Programmzeile beendet wird. Sind mehrere Programme im Hauptspeicher gespeichert, wird bei fehlender END-Anweisung das nächste Programm sofort gestartet.

Beispiel:

```
10 PRINT "START PRO 1"
20 FOR I = 1 TO 20
30 PAUSE I
40 NEXT I
100 PRINT "START PRO 2"
110 PRINT "PRO 1 KEIN END"
120 END
```

Ausgabe nach RUN:

START PRO 1

ENTER

1.
2.
.
.
.
20.

START PRO 2

ENTER

PRO 1 KEIN END

EXP

Funktion:

Exponentialfunktion e^X.

Syntax:

EXP numerischer Ausdruck

Bemerkungen:

Die Basiszahl e ist als EXP 1 = 2.718281828 gespeichert.

Beispiel:

```
10 A = LN 100
20 B = EXP A
30 PRINT B
40 END
```

Ausgabe nach RUN:

```
            100.
```

FAC

Funktion:

Berechnet die Fakultät eines numerischen Ausdrucks.

Syntax:

FAC numerischer Ausdruck

Bemerkungen:

Die Fakultät eines numerischen Ausdrucks berechnet sich nach der folgenden Formel:

Fakultät von X = 1*2*3*....*X

Beispiel:

```
10 A = 5
20 B = FAC A
30 PRINT B
40 END
```

Ausgabe nach RUN:

```
            120.

1 * 2 * 3 * 4 * 5 = 120
```

FOR...TO...STEP - NEXT

Funktion:

Wiederholung von Anweisungen in einer Schleife.

Syntax:

FOR num. Variable = num Ausdruck **TO** num. Audruck
FOR num. Variable = num. Ausdruck **TO** num. Audruck STEP num. Ausdruck

Bemerkungen:

Eine Programmschleife besteht aus einer FOR-Anweisung, den nachfolgenden Programmanweisungen und der NEXT-Anweisung.

Die FOR-Anweisung bewirkt, daß der von FOR und NEXT eingeschlossene Programmteil in Abhängigkeit der numerischen Ausdrücke (Anfangswert, Endwert, Schrittweite) wiederholt ausgeführt wird.

Wird die Schrittweite (STEP) nicht definiert, so ist sie automatisch auf den Wert 1 festgesetzt.

Ist der Endwert der Schleife erreicht, so wird die Programmausführung nach der NEXT-Anweisung fortgesetzt.

Programmschleifen können bis zu fünf Ebenen ineinander geschachtelt sein. Es ist nur darauf zu achten, daß immer zuerst die innere Schleife mit NEXT abgeschlossen werden muß.

Beispiel:

```
60 FOR I = 1 TO 10
70 FOR J = I + A TO 10
80 FOR K = C * D TO 30

120 NEXT K
130 NEXT J
140 NEXT I
```

RICHTIGE SCHACHTELUNG

```
120 FOR A = 0 TO 10
130 FOR C = A * 2 TO B

180 NEXT A
200 NEXT C
```

FALSCHE SCHACHTELUNG

Eine Sprunganweisung (GOTO, GOSUB) im Schleifenbereich kann während der Ausführung der Schleife in einen Programmteil außerhalb der Schleife verzweigen.
Es darf jedoch nicht von außerhalb in eine Programmschleife verzweigt werden. Eine Ausnahme ist der Rücksprung in eine Schleife aus einem Unterprogramm. Wurde ein solches während des Schleifendurchlaufs durch GOSUB aufgerufen, lenkt die Programmausführung nach Abarbeiten des Unterprogramms in die Schleife zurück.

Ein unzulässiger Sprung in eine FOR...NEXT-Schleife führt bei der folgenden NEXT-Anweisung zu der Fehlermeldung ERROR 5.

```
1 GOTO 9
5 FOR I = 1 TO 10                    FALSCHE
9 PAUSE "FALSCH"                     VERZWEIGUNG
10 NEXT I


100 FOR I = 1 TO 100
110 READ A$
120 IF A$ = "ENDE" GOTO 190          RICHTIGE
                                     VERZWEIGUNG
180 NEXT I
190 END


10 FOR I = 1 TO 100 STEP 5
20 READ A$
30 GOSUB "KASSETTE"                  RICHTIGE
40 .....                             VERZWEIGUNG
100 NEXT I
500 "KASSETTE": PRINT#"DATEN";A$
600 RETURN
```

Vor dem ersten Durchlauf wird der Schleifenvariablen der Wert vom ersten numerischen Ausdruck (Anfangswert) der FOR-Anweisung zugewiesen.
Erreicht der Programmablauf die NEXT-Anweisung, so wird die Schrittweite (dritter numerischer Ausdruck) zum Wert der Schleifenvariablen addiert und der Programmteil mit diesem Wert erneut durchlaufen. Dies wird solange fortgesetzt, so lange der Wert innerhalb der im folgenden angegebenen Bereiche liegt:

(1) Bei positiver Schrittweite:

Anfangswert $<$ Schleifenvariable $<$ Endwert + Schrittweite

(2) oder bei negativer Schrittweite:

Schrittweite + Endwert $<$ Schleifenvariable $<$ Anfangswert

Der Wert der Schleifenvariablen einer Sprunganweisung darf im Bereich zwischen -9.999999999E99 und 9.999999999E99 liegen. Für die Schrittweite gilt der gleiche Bereich; wird der Wert 0 gewählt, ist die Schleife unendlich.

Nach Beendigung der Schleifenanweisung bleibt die Schleifenvariable definiert und behält den ihr zuletzt zugewiesenen Wert.

Die Werte der zugewiesenen Anfangs-, End- und Schrittweitenparameter bleiben fest, solange die Schleife durchlaufen wird. Das gilt auch, wenn der Wert einer zu ihrer Berechnung verwendeten Variablen innerhalb oder außerhalb der Schleife geändert wird.

Beispiele:

zu (1) Einlesen der Zeilen einer Matrix B=B(I,J) über eine geschachtelte Schleifenanweisung.

```
(1)  100 FOR I = 1 TO N
     110 FOR J = 1 TO N
     120 READ B (I,J)
     130 NEXT J
     140 NEXT I
```

```
(2)  10 PAUSE"LISTE DER QUADRATZAHLEN"
     20 PAUSE"VON 10 BIS 20"
     30 FOR Z = 10 TO 20
     40 Q = SQU Z
     50 PAUSE Q
     60 NEXT Z
     70 END
```

GOSUB

Funktion:

Verzweigung zu einem Unterprogramm.

Syntax:

GOSUB numerischer Ausdruck
GOSUB Textausdruck

Bemerkungen:

Werden in einem Programm an verschiedenen Stellen gleiche Befehlsfolgen verwendet, so werden diese als Unterprogramm (Subroutine) geschrieben.

Die GOSUB-Anweisung verzweigt den Programmablauf zu der Programmzeile, in der das entsprechende Unterprogramm beginnt. Analog zur GOTO-Anweisung wird das Sprungziel entweder als Zeilennummer oder als Markenname angegeben.

Ein Unterprogramm wird mit einer RETURN-Anweisung abgeschlossen; der Programmablauf wird dann mit der der GOSUB-Anweisung folgenden Anweisung fortgesetzt.

Die GOSUB-Anweisung kann an beliebiger Stelle im Programm stehen.

Bis zu zehn GOSUB-Anweisungen können ineinander geschachtelt werden. Dabei darf jedoch z.B. ein Unterprogramm A, das ein Unterprogramm B aufruft, nicht wiederum von B aufgerufen werden. Anderseits kann B sowohl vom Hauptteil des Programms als auch von A aus aufgerufen werden.

Beispiel:

(1)
```
120 GOSUB 300
130 PRINT "A="; A
.
.
180 GOSUB 300
190 IF A = 10 THEN 250
.
.
300 LET X = C + B        }
.                        } Unterprogramm
.
380 RETURN               }
```

(2)

```
    Hauptprogramm              Unterprogramme

                               1. Ebene              2. Ebene

50 GOSUB "SIMP"            400 "SIMP":           500 LET X = A * B
60                         450 GOSUB 500
.                          .                     .
.                          .                     .
.                          .                     590 RETURN
160 GOSUB "SIMP"           490 RETURN
170
.
.
.
270 GOSUB "SON"            700 "SON":
280                        .
.                          .
.                          .
.                          790 RETURN
390 END
```

In diesem Beispiel werden mehrere Unterprogramme ineinander verschachtelt. In Zeile 50 und 160 wird das Unterprogramm "SIMP" aufgerufen. Dieses wiederum ruft in Zeile 450 das Unterprogramm der zweiten Ebene in Zeile 500 auf. Der Rücksprung erfolgt jeweils zu der Anweisung, die auf den entsprechenden GOSUB-Aufruf folgt.

GOTO

Funktion:

Verzweigt die Programmausführung.

Syntax:

```
GOTO num. Ausdruck < Zeilennummer >
GOTO Textausdruck < Markenname >
```

Bemerkungen:

Die GOTO-Anweisung verzweigt den Programmablauf zu einer bestimmten Programmzeile, die entweder als numerischer Ausdruck (Zeilennummer) oder als Textausdruck (Markenname) angegeben sein kann.

Ist die angegebene Sprungadresse im Programm nicht vorhanden, so erfolgt die Fehlermeldung ERROR 4.

Beispiel:

```
10 A = 1
20 "LABEL-1": PRINT USING"#####";A;A ^ 2;A ^ 3
30 A = A + 1
40 IF A > 10 THEN GOTO 60
50 GOTO "LABEL-1"
60 END
```

Dieses Programm berechnet die Quadrat- und Kubikzahlen zwischen 1 und 10.

Ergebnis:

```
 1    1    1
 2    4    8
 3    9   27
 4   16   64
 5   25  125
 6   36  216
 7   49  343
 8   64  512
 9   81  729
10  100 1000
```

HCS

Funktion:

Berechnet den Hyperbelcosinus eines numerischen Ausdrucks.

Syntax:

HCS numerischer Ausdruck

Beispiel:

```
10 INPUT X
20 LET Y = HCS X
30 PRINT"X=";X;" ";"HCS X=";Y
```

Ausgabe nach RUN:

```
?                          1  ENTER

X=1. HCS X=1.543
```

HSN

Funktion:

Berechnet den Hyperbelsinus eines numerischen Ausdrucks.

Syntax:

HSN numerischer Ausdruck

Beispiel:

```
10 INPUT X
20 LET Y = HSN X
30 PRINT"X=";X;" ";"HSN X=";Y
```

Ausgabe nach RUN:

```
?                          1 [ENTER]

X=1. HSN X=1.175
```

HTN

Funktion:

Berechnet den Hyperbeltangens eines numerischen Ausdrucks.

Syntax:

HTN numerischer Ausdruck

Beispiel:

```
10 INPUT X
20 LET Y = HTN X
30 PRINT"X=";X;" ";"HTN X=";Y
```

Ausgabe nach RUN:

```
?                          1  ENTER

X=1. HTN X=0.761
```

IF...THEN

Funktion:

Programmverzweigung in Abhängigkeit von einem logischen Ausdruck.

Syntax:

IF Vergleichsausdruck **THEN** Anweisung
IF Vergleichsausdruck **THEN** num. Ausdruck
IF Vergleichsausdruck **THEN** Textausdruck

IF num. Ausdruck **THEN** Anweisung
IF num. Ausdruck **THEN** num. Ausdruck
IF num. Ausdruck **THEN** Textausdruck

Bemerkungen:

Der Vergleichsausdruck bzw. der numerische Ausdruck stellt die Bedingung dar, die auf 'wahr' oder 'falsch' geprüft wird.

Im Fall des numerischen Ausdrucks wird die Bedingung als 'wahr' angesehen, wenn der Wert größer (>) 0, und als 'falsch' wenn er kleiner gleich (≤) 0 ist.

Der Vergleichsausdruck wird nach den mathematischen Regeln der Operatoren <, <=, =, >=, >, <> bestimmt und liefert die Werte "1" (wahr) oder "0" (falsch).

Bei der Anwendung der Operatoren auf Textausdrücke wird auf lexikalische Reihenfolge geprüft. Dabei werden nur die ersten 16 Zeichen des Textausdruckes berücksichtigt.

Ist die Bedingung einer IF-Abfrage 'wahr', so werden die auf THEN folgenden Anweisungen ausgeführt.

Wird nach einer THEN-Anweisung nur eine Zeilennummer oder ein Textausdruck angegeben, so stellt dies die verkürzte Schreibweise für GOTO dar.

Beispiel: IF A < 3 THEN 100 = IF A < 3 THEN GOTO 100

Anweisungen, die in der gleichen Zeile stehen wie die IF...THEN-Anweisung, werden nur ausgeführt, wenn das Ergebnis der Abfrage 'wahr' ist.

Mit dem **PC-1401/1402** können Rechnungen mit bis zu 12stelliger Mantisse durchgeführt werden. Die Mantisse wird intern bis zur 12. Stelle berechnet, für die Anzeige wird jedoch ab der 10. Stelle gerundet.

Beispiel:

	internes Ergebnis	Anzeige
5/9 ---→	5.55555555555E-01	5.555555556E-01
5/9*9 ---→	4.99999999999E-00	5.

In dieser Weise werden Rechnungen bis zu 12 Stellen ausgeführt. Daher kann es zu Differenzen im Ergebnis von Rechnungen kommen, wenn man sie verkettet oder getrennt ausführt.

Beispiel:

	Ergebnis
a.) 1/6 * 3	0,5
b.) 1/6 [ENTER] *3	5.000000001E-01

Obgleich die Ausgangspunkte und die Berechnungen gleich sind, werden zwei unterschiedliche Ergebnisse errechnet, weil bei der Berechnung von

a.) einmal, und bei der Berechnung von b.) zweimal gerundet wird.

In einer IF-Anweisung kann diese Differenz dazu führen, daß das Programm nicht mehr fehlerfrei arbeitet.

```
300:  "A":INPUT A
310:  X=3
320:  Y=1/A
330:  Z=Y*X
340:  IF Z=1/A*X THEN 370
350:  PRINT "KEINE VERZWEI
      G."
360:  GOTO 300
370:  PRINT "VERZWEIGUNG"
380:  GOTO 300
```

Während bei der Eingabe von z. B. "3" die Verzweigung im nebenstehenden Programm ordnungsgemäß ausgeführt wird, führt die Eingabe von z. B. "6" zu einer falschen Abzweigung.

Dieser Fehler kann vermieden werden, wenn in der Zeile 340 der Vergleichsausdruck IF Z = 1/A * X geändert wird in

IF Z = Y * X

Beispiele:

(1) 10 IF A = 10 THEN PRINT "A = 10"

Wenn der Wert der Variablen A = 10 ist, wird am Display "A = 10" angezeigt.

(2) 10 IF A$ = "JA" THEN GOTO 200

(3) 10 IF A$ = "JA" GOTO 200

(4) 10 IF A$ = "JA" THEN 200

Die Beispiele 2,3 und 4 haben dieselbe Wirkung. Die Programmzeile 10 enthält die Abfrage für die Textvariable A$. Ist A$ ="JA", so verzweigt das Programm zu Zeile 200.

(5)
```
10 X = 100
 .
 .
 .
50 IF X THEN 200
 .
 .
 .
200 PRINT "SHARP"
```

Der Programmverlauf wird in Zeile 50 zu Zeile 200 verzweigt, da die Abfrage IF X > 0 ergibt und daher "wahr" ist.

(6)

```
10 A = RND 500
20 INPUT "ZUF.ZAHL ?";B
30 IF A < B GOTO 110
40 IF A > B GOTO 100
50 IF A = B PRINT "RICHTIG (ENTER)": GOTO 200
100 PAUSE "ZAHL I. GROESSER": GOTO 20
110 PAUSE "ZAHL I. KLEINER": GOTO 20
200 INPUT "NOCHMAL-J/N"; C$
210 IF C$ = "J" THEN 10
220 IF C$ <> "N" THEN 200
230 END
```

INKEY$

Funktion:

Tastaturabfrage

Syntax:

Textvariable = **INKEY$**

Bemerkungen:

INKEY$ fragt die Tastatur ab und ergibt, je nach gedrückter Taste einen Textausdruck von einem Zeichen Länge oder einen leeren Textausdruck (ASCII-Null), wenn keine Taste gedrückt wurde.

Beispiel:

```
10 A$ =""
20 A$ = INKEY$
30 IF A$ = "E" THEN 900
40 IF A$ = "S" THEN 60
50 GOTO 20
60 PRINT A$;"START (PROGRAMM)"
.
.
.
700 GOTO 20
.
.
.
900 PRINT "ENDE (PROGRAMM)"
910 END
```

Wird bei der Abarbeitung der Programmzeile 20 die Taste [S] oder [E] gedrückt, verzweigt sich das Programm gemäß den Anweisungen in Zeile 30 und 40. Wird keine oder eine andere Taste gedrückt, springt die Programmausführung in Zeile 50 zurück zu Zeile 20.
Mit der Taste [S] wird das Programm gestartet, mit der Taste [E] beendet. Nach jedem Durchlauf muß es mit [S] wieder gestartet werden.

INPUT

Funktion:

Weist Variablen Werte zu, die über die Tastatur eingegeben werden.

Syntax:

INPUT numerische Variable
INPUT Textvariable
INPUT"Textausdruck", numerische Variable
INPUT"Textausdruck"; numerische Variable
INPUT"Textausdruck", Textvariable
INPUT"Textausdruck"; Textvariable

Bemerkungen:

Die Ausführung des Programms wird unterbrochen, und auf der Anzeige erscheint der eingegebenen Textausdruck oder ein ? (Fragezeichen). Das Fragezeichen dient als Hinweis darauf, daß die Werte für die Variablen über die Tastatur eingegeben werden können. Jede einzelne Eingabe wird mit [ENTER] abgeschlossen. Dies wird so lange wiederholt, bis für alle Variablen die Werte eingegeben sind. Im Anschluß daran wird die Programmausführung fortgesetzt.

Beispiel:

```
(1)  10 INPUT A
     20 IF A = 1234 THEN 100
     30 GOTO 10
     100 PRINT A
```

Ausgabe nach RUN:

?

Eingabe: 1 2 3 4

1234 [ENTER]

Anzeige nach Abarbeitung der Programmzeile 100:

1234.

Ist die eingegebene Zahlenreihe nicht 1234, so springt die Programmausführung von Zeile 30 zu Zeile 10 zurück.

(2)
```
10 INPUT A$
20 IF A$ = "START" THEN 100
30 GOTO 10
100 PRINT "PROG.ENDE"
```

Ausgabe nach RUN:

?

Eingabe: START

START [ENTER]

Anzeige nach Abarbeitung der Programmzeile 100:

PROG.ENDE

(3)
```
10 INPUT A,B
20 IF A > B THEN 100
30 GOTO 10
100 PRINT "PROG.ENDE"
```

Ausgabe nach RUN:

?

Eingabe: 1 und 2 (Wertzuweisung für Variable A)

12 [ENTER]

?

Eingabe: 9 (Wertzuweisung für Variable B)

9 [ENTER]

Anzeige nach Abarbeitung der Programmzeile 100:

PROG.ENDE

(4)
```
10 INPUT"DATA-1",A
20 IF A > 12 THEN 100
30 GOTO 10
100 PRINT "PROG.ENDE"
```

Ausgabe nach RUN:

DATA-1

Eingabe: 123 (Wertzuweisung für Variable A)

123 ENTER

Anzeige nach Abarbeitung der Programmzeile 100:

PROG.ENDE

(5)
```
10 INPUT "DATA-1=";A,"DATA-2=";B
20 IF A > B THEN 100
30 GOTO 10
100 PRINT"PROG.ENDE"
```

Ausgabe nach RUN:

DATA-1=_

Eingabe: 75

DATA-1=75_ ENTER

DATA-2=_

Eingabe: 71

DATA-2=71_ ENTER

Anzeige nach Abarbeitung der Programmzeile 100:

PROG.ENDE

INPUT# (nur mit Option CE-126P)

Funktion:

Daten werden vom Band gelesen und Variablen zugewiesen.

Syntax:

INPUT# Variable
INPUT# Textkonstante; Variable

Bemerkungen:

Mit der INPUT#-Anweisung kann man Daten, die mit der PRINT#-Anweisung auf Band gespeichert wurden, lesen und sie Variablen zuweisen.

Die Sprachelemente haben dieselbe Bedeutung wie in der PRINT#-Anweisung. Die Textkonstante bezeichnet den Namen des zu lesenden Datensatzes. Der einzelnen Variablen oder einer Folge von Variablen und Felder werden die Werte zugewiesen, die mit der PRINT#-Anweisung abgespeichert wurden. Felder sind identisch der PRINT#-Anweisung anzugeben.

Wenn die Anzahl der einzulesenden Variablen nicht mit der Anzahl der gespeicherten Variablen übereinstimmt, so ergeben sich folgende Zustände:

1. Anzahl der einzulesenden Variablen ist größer als die Anzahl der auf Band gespeicherten Variablen:

   Die Übertragung wird nicht beendet und muß mit BREAK abgebrochen werden oder es wird die Fehlermeldung ERROR 8 angezeigt.

2. Die Anzahl der einzulesenden Variablen ist kleiner als die Anzahl der auf Band gespeicherten Variablen:

   Die überzählige Variable wird ignoriert.

Beispiel:

```
INPUT# A, K*, A1$(*)
INPUT#"DATEI"; A*, X1(*)
```

Hinweis:

Bei korrektem Einlesen der Daten wird in der rechten Stelle der Anzeige ein Sternsymbol angezeigt.

INT

Funktion:

Die INTeger-Funktion ermittelt den ganzzahligen Anteil eines numerischen Ausdrucks.

Syntax:

INT numerischer Ausdruck

Bemerkungen:

Ist der Wert des numerischen Ausdrucks positiv, so ergibt die INT-Funktion einen Wert, der gleich oder kleiner dem Wert des numerischen Ausdrucks ist. Ist der Wert jedoch negativ, so ergibt sich ein Wert, dessen Betrag gleich oder größer dem Wert des numerischen Ausdrucks ist.

Beispiel:

```
10 A = INT (PI)
20 B = INT (3.99)
30 C = INT (-3.14)
40 PRINT A;" ";B;" ";C
50 END
```

Ausgabe nach RUN:

```
3.  3.  -4.
```

LEFT$

Funktion:

Ergibt die ersten n Zeichen eines Textausdrucks

Syntax:

LEFT$ (Textausdruck, numerischer Ausdruck)

Bemerkungen:

Der Wert des numerischen Ausdrucks bestimmt die Anzahl der Zeichen, die von links aus der Zeichenkette des Textausdrucks entnommen werden und die die Ergebniszeichenfolge bilden.

Beispiele:

```
5 DIM S$(0)*13
10 S$(0) = "SHARP HAMBURG"
20 T$ = LEFT$ (S$(0), 5)
30 PRINT T$
40 END
```

Ausgabe nach RUN:

SHARP

In Programmzeile 20 werden dem Textausdruck S$(0) die ersten 5 Zeichen (SHARP) entnommen und der Variablen T$ zugewiesen.

```
5 DIM P$(0)*13
10 P$(0) = "8000 MUENCHEN"
20 T$ = LEFT$ (P$(0), 4)
30 PRINT "PLZ", T$
40 END
```

Ausgabe nach RUN:

PLZ8000

In Programmzeile 20 werden dem Textausdruck P$(0) die ersten 4 Zeichen (8000) entnommen und der Variablen T$ zugewiesen.

LEN

Funktion:

Berechnet die Anzahl der Zeichen eines Textausdrucks.

Syntax:

LEN Textausdruck

Bemerkungen:

Leerstellen werden bei der Berechnung der Zeichen mitgerechnet.

Beispiel:

```
10 A = LEN "SHARP"
20 B$ = "HAMBURG"
30 C = LEN B$
40 PRINT A;" ";C
50 END
```

Ausgabe nach RUN:

```
5. 7.
```

5 ist die Anzahl der Zeichen des Textausdrucks "SHARP", 7 die von "HAMBURG".

LET

Funktion:

Weist einer Variablen (einfach oder indiziert) einen Wert zu.

Syntax:

LET numerische Variable = numerische Variable
LET numerische Variable = numerischer Ausdruck
LET Textvariable = Textvariable
LET Textvariable = Textausdruck

Bemerkungen:

Die LET-Anweisung weist einer Variablen (numerisch oder Text) einen Wert zu. Dabei muß der Variablentyp dem zugewiesenen Wert entsprechen, d.h., einer numerischen Variablen darf nur ein numerischer Wert zugewiesen werden, einer Textvariablen nur ein Textausdruck.
Zur Umwandlung einer numerischen Variablen in eine Textvariable und umgekehrt können die Anweisungen STR$ bzw. VAL verwendet werden.

Beispiel:

(1) LET A = 5

(2) LET X = Y

(3) LET B$ = "SHARP"

(4) LET C$ = D$

LIST

Funktion:

Auflisten der Programmzeilen.

Syntax:

LIST
LIST num. Ausdruck < Zeilennummer >
LIST num. Ausdruck, num. Ausdruck < Bereich >

Bemerkungen:

Die Ausführung des LIST-Kommandos kann nur im PRO-Mode erfolgen. Im RUN-Mode wird die Fehlermeldung ERROR 9 angezeigt.

LIST ohne zusätzliche Angabe listet die Programmzeile mit der niedrigsten Zeilennummer auf. Durch Angabe einer bestimmten Zeilennummer oder eines Markennamens lassen sich bestimmte Programmzeilen zur Anzeige bringen. Ist die angegebene Programmzeile nicht vorhanden, so wird die nächsthöhere gelistet.

Durch Betätigung der Tasten [↑] oder [↓] wird die folgende bzw. die vorangehende Programmzeile angezeigt.

Beispiel:

(1) LIST 100

Die Programmzeile 100 wird zur Anzeige gebracht.

(2) LIST

(3) LIST "S"

LLIST (nur mit Option CE-126P)

Funktion:

Auflisten der Programmzeilen auf dem Drucker.

Syntax:

LLIST
LLIST numerischer Ausdruck < Zeilennummer >
LLIST numerischer Ausdruck , numerischer Ausdruck < Bereich >
LLIST numerischer Ausdruck , 1. zu druckende Zeile
LLIST numerischer Ausdruck

Bemerkungen:

Die LLIST-Anweisung hat die gleiche Funktion wie die LIST-Anweisung. Das Programm wird jedoch auf dem Drucker und nicht auf der Anzeige gelistet.
Ohne Angaben von Programmzeilen wird das gesamte Programm ausgedruckt.
Existieren die angegebenen Zeilennummern nicht, so wird das Listing bei der nächsthöheren Zeilennummer begonnen bzw. beendet.

Beispiel:

LLIST

Das im Hauptspeicher gespeicherte Programm wird am Drucker gelistet.

LLIST 10,100

Die Programmzeilen beginnend, mit der Zeilennummer 10 bis zur Zeilennummer 100, werden am Drucker gelistet.

LLIST 100,

Das im Hauptspeicher gespeicherte Programm wird ab Zeile 100 am Drucker gelistet.

LLIST, 100

Das im Hauptspeicher gespeicherte Programm wird bis Zeile 100 am Drucker gelistet.

LPRINT (nur mit Option **CE-126P**)

Funktion:

Numerische Werte und Textausdrücke am Drucker ausgeben.

Syntax:

LPRINT numerischer Ausdruck
LPRINT Textausdruck
LPRINT numerischer Ausdruck,(;) numerischer Ausdruck
LPRINT Textausdruck,(;) Textausdruck
LPRINT numerischer Ausdruck,(;) Textausdruck
LPRINT Textausdruck,(;) numerischer Ausdruck
LPRINT USING Format; numerischer Ausdruck; ...

Bemerkungen:

Der USING-Teil der LPRINT-Anweisung spezifiziert das Ausgabeformat, wie unter USING beschrieben.

Mit dem Komma wird die Aufteilung der 24spaltigen Druckzeile in zwei gleich große Felder gesteuert.

Das Semikolon trennt die einzelnen Elemente einer Liste von Ausdrücken und steuert die Position des Cursors.

Wird die Cursor-Positionierung nicht explizit durch die Verwendung eines Semikolons vereinbart, so werden numerische Ausdrücke rechtsbündig und Textausdrücke linksbündig gedruckt.

Nach der Ausführung einer LPRINT-Anweisung wird der Programmablauf, im Gegensatz zur PRINT-Anweisung, automatisch fortgesetzt.

Hinweise:

1. Werden mehr als 24 Zeichen ausgegeben, wird zwei- oder mehrzeilig gedruckt. Für den Fall, daß die Ausdrücke der LPRINT-Liste mit einem Komma verknüpft sind, werden Textausdrücke auf 12 Zeichen begrenzt.

2. Wird in einer LPRINT-Anweisung die USING-Anweisung verwendet, so ist das festgelegte Format für alle weiteren LPRINT-Anweisungen bis zur nächsten USING-Anweisung gültig.

3. Das USING-Format wird durch die RUN-Anweisung oder durch Drücken der Tasten [SHIFT] und [CA / C·CE] gelöscht.

Die einzelnen Formate der LPRINT-Anweisung haben folgende Wirkung:

LPRINT numerischer Ausdruck
LPRINT Textausdruck

Der in der LPRINT-Anweisung spezifizierte Inhalt wird rechts- bzw. linksbündig ausgedruckt.

LPRINT numerischer Ausdruck, numerischer Ausdruck
LPRINT Textausdruck, Textausdruck

Die Zeile wird in zwei Felder geteilt. Der erste numerische Ausdruck (bzw. Textausdruck) wird in der linken, der zweite in der rechten Hälfte jeweils rechts- bzw. linksbündig gedruckt.

LPRINT numerischer Ausdruck ; numerischer Ausdruck
LPRINT Textausdruck ; Textausdruck

Der Inhalt der LPRINT-Liste wird, beginnend mit der jeweiligen Cursor-Position, ausgedruckt.

Beispiele:

(1)
```
10 A = 3.1415
20 B$ = "SHARP"
30 LPRINT A
40 LPRINT B$
```

Ausgabe nach RUN:

```
                3.1415
SHARP
```

(2)
```
10 A = 3.1415
20 B$ = "SHARP"
30 C = 7.89
40 D$ = "PC-1401"
50 LPRINT A;B$
60 LPRINT B$,A
70 LPRINT B$,D$
80 LPRINT A,C
```

Ausgabe nach RUN:

```
3.1415SHARP
SHARP           3.1415
SHARP       PC-1401
     3.1415        7.89
```

(3)
```
10 A = 3.1415
20 B$ = "SHARP"
30 C = 7.89
40 D$ = "PC-1401"
50 LPRINT A;B$;D$;C
60 LPRINT A;" ";B$;" ";D$;" ";C
```

Ausgabe nach RUN:

```
3.1415SHARPPC-14017.89
3.1415 SHARP PC-1401 7.8
9
```

(4)
```
10 A = 3.1415
20 B$ = "SHARP"
30 C = 7.89
40 D$ = "PC-1401"
50 E$ = "&&###.#"
60 F$ = "###.#&&"
70 LPRINT USING E$; B$;A
80 LPRINT USING F$; B$;A
90 LPRINT USING; B$;A
100 LPRINT USING E$; A, C
110 LPRINT USING F$; B$,D$
120 LPRINT USING "##.####";A,C
130 LPRINT USING; A,C
```

Ausgabe nach RUN:

```
SH  3.1
SH  3.1
SHARP3.1415
         3.1          7.8
SH           PC
      3.1415       7.8900
      3.1415         7.89
```

LN

Funktion:

Berechnet den natürlichen Logarithmus eines numerischen Ausdrucks.

Syntax:

LN numerischer Ausdruck

Bemerkungen:

Der Wert des numerischen Ausdrucks darf nicht negativ sein.

Beispiel:

```
10 A = LN 100
20 PRINT A
30 END
```

Ausgabe nach RUN:

```
4.605170186
```

LOG

Funktion:

Berechnet den Zehnerlogarithmus eines numerischen Ausdrucks.

Syntax:

LOG numerischer Ausdruck

Bemerkungen:

Der Wert des numerischen Ausdrucks darf nicht negativ sein.

Beispiel:

```
10 A = LOG 100
20 PRINT A
30 END
```

Ausgabe nach RUN:

```
              2.
```

MEM

Funktion:

Ermittelt den freien Speicherbereich des Hauptspeichers.

Syntax:

MEM

Bemerkungen:

MEM gibt die Anzahl der unbelegten Bytes im Hauptspeicher an. Berücksichtigt werden dabei sowohl der Programm- als auch der Feldvariablenspeicher.

Beispiel:

MEM

Anzeige am Display:

3534 (PC-1401)
9678 (PC-1402)

Ist das Programm durch ein PASS-Wort geschützt, so kann es durch die NEW-Anweisung gelöscht werden, die Standardvariablen werden auf Null gesetzt (s. PASS).

MID$

Funktion:

Ergibt den mittleren Teil eines Textausdrucks mit der Zeichenzahl n ab dem p-ten Zeichen.

Syntax:

MID$ (Textausdruck, numerischer Ausdruck 1, numerischer Ausdruck 2)

Bemerkungen:

Der Wert des ersten numerischen Ausdrucks legt die Position innerhalb des Textausdrucks fest, von der ab die Zeichen entnommen werden. Der Wert des zweiten numerischen Ausdrucks bestimmt die Zeichenanzahl, die entnommen werden soll.

Beispiel:

```
10 DIM A$(0)*20
20 A$(0)="SHARP HAMBURG"
30 B$=MID$(A$(0),7,4)
40 PRINT"2000 ";B$
50 END
```

Ausgabe nach RUN:

```
2000 HAMB
```

Dem Textausdruck "SHARP HAMBURG" werden, beginnend mit dem siebten Zeichen, vier Zeichen entnommen und der Textvariablen B$ zugewiesen.

NEW

Funktion:

Löscht den Hauptspeicher.

Syntax:

NEW

Bemerkungen:

Der Hauptspeicher wird gelöscht. Alle Variablen werden auf Null gesetzt.

Die NEW-Anweisung kann nur im PRO-Mode eingegeben werden, ansonsten wird die Fehlermeldung ERROR 9 angezeigt.

Beispiel:

NEW

NOT

Funktion:

Invertieren der Binärdarstellung eines numerischen Ausdrucks.

Syntax:

NOT numerischer Ausdruck

Bemerkungen:

Bei der "NOT"-Funktion wird der numerische Ausdruck als 16stellige Binärzahl interpretiert. Der Wert des numerischen Ausdrucks muß dabei im Bereich zwischen -32768 und +32767 liegen, ansonsten erfolgt die Fehlermeldung ERROR 3.
Die "NOT"-Funktion invertiert jede Stelle des in Binärformat dargestellten Ausdrucks.
Negative Binärzahlen werden durch das Zweierkomplement dargestellt, d.h. -X ergibt (NOT X) + 1.

Beispiel:

X (dezimal)	X (binär)	NOT X (binär)	NOT X (dezimal)
-1	11111111 11111111	00000000 00000000	0
0	00000000 00000000	11111111 11111111	-1
1	00000000 00000001	11111111 11111110	-2
15	00000000 00001111	11111111 11110000	-16

Hinweis: Bitte beachten Sie, daß bei der NOT-Funktion die Hierarchie der mathematischen Operationen nicht gültig ist:

```
A = 2
B = 3
C = A < B

NOT A < B    ist 1 (mathematisch falsch)
NOT C        ist -2
```

Durch Klammersetzung erreicht man ein mathematisch richtiges Ergebnis: NOT (A < B)

ON...GOTO

Funktion:

Verzweigt die Programmausführung in Abhängigkeit von dem Wert eines numerischen Ausdrucks.

Syntax:

ON Sprungindex **GOTO** numerischer Ausdruck, numerischer Ausdruck
ON Sprungindex **GOTO** Textausdruck, Textausdruck

Bemerkungen:

Durch den ganzzahligen positiven Wert des Sprungindex wird einer der GOTO-Anweisung folgenden numerischen Ausdrücke bzw. Textausdrücke (Sprungzielliste) ausgewählt.

Hat der Sprungindex den Wert 1, verzweigt das Programm zum ersten Element der Sprungzielliste, beim Wert 2 zum zweiten Element usw., beim Wert i zum i-ten Element.

Ist der Wert des Sprungindex $<= 0$, so verzweigt das Programm nicht, und die Programmausführung wird mit der der ON...GOTO-Anweisung folgenden Anweisung fortgesetzt.

Beispiele:

(1)
```
40 FOR I = 0 TO 2

50 ON I GOTO 100, 200
60 PRINT "KEINE VERZW."
100 PRINT "ZEILE 100"
200 PRINT "ZEILE 200"
400 NEXT I
```

In diesem Beispiel wird die Schleife dreimal durchlaufen (I = 0, 1, 2). Hat der Sprungindex den Wert 0, wird das Programm mit Zeile 60 fortgesetzt. Ansonsten verzweigt das Programm jeweils zu der dem aktuellen Sprungindex entsprechenden Zeilennummer.

Ausgabe nach RUN:

```
KEINE VERZW.
                    ENTER
ZEILE 100
                    ENTER
ZEILE 200
```

(2)
```
5 USING "##":  B$= " "
10 FOR I = -2 TO 5
20 ON I GOTO 50,30,60,70
30 LPRINT I;B$;"KEIN SPRUNG"
40 GOTO 200
50 LPRINT I;B$;"SPRUNG NACH 50"
55 GOTO 200
60 LPRINT I;B$;"SPRUNG NACH 60"
65 GOTO 200
70 LPRINT I;B$;"SPRUNG NACH 70"
200 NEXT I
```

In diesem Beispiel kann der Sprungindex die Werte -2,-1,0,1,2,3,4,5 annehmen. Für die Werte < 0 (-2,-1,0) erfolgt kein Sprung; der Programmablauf wird mit Zeile 30 fortgesetzt.
Nimmt I den Wert 1,2,3 oder 4 an, so wird je nach dem aktuellen Wert von I zum ersten, zweiten, dritten oder vierten Element der Sprungzielliste verzweigt.
Für I = 5 gibt es kein Sprungziel; die Programmausführung wird mit Zeile 30 fortgesetzt.

Ausgabe nach RUN mit Drucker (Option **CE-126P**):

```
-2  KEIN SPRUNG
-1  KEIN SPRUNG
 0  KEIN SPRUNG
 1  SPRUNG NACH 50
 2  KEIN SPRUNG
 3  SPRUNG NACH 60
 4  SPRUNG NACH 70
 5  KEIN SPRUNG
```

ON...GOSUB

Funktion:

Verzweigt die Programmausführung zu einem Unterprogramm in Abhängigkeit vom Wert eines numerischen Ausdrucks.

Syntax:

ON Sprungindex **GOSUB** numerischer Ausdruck, numerischer Ausdruck
ON Sprungindex **GOSUB** Textausdruck, Textausdruck

Bemerkungen:

Durch den ganzzahligen positiven Wert des Sprungindex wird einer der GOSUB-Anweisung folgenden numerischen Ausdrücke bzw. Textausdrücke (Sprungzielliste) ausgewählt.

Hat der Sprungindex den Wert 1, verzweigt das Programm zum ersten Element der Sprungzielliste, beim Wert 2 zum zweiten Element usw., beim Wert i zum i-ten Element.

Ist der Wert des Sprungindex $<= 0$, so verzweigt das Programm nicht, und die Programmausführung wird mit der der ON...GOSUB-Anweisung folgenden Anweisung fortgesetzt.

Beispiel:

```
10 INPUT D
20 ON (SGN (D)+2) GOSUB "MINUS", "NULL", "PLUS"
100 "MINUS"
200 "NULL"
300 "PLUS"
```

SGN (D) kann die Werte -1,0,1 annehmen, jeweils für $D < 0$, $D = 0$, $D > 0$. Damit erhält SGN (D)+2 die Werte 1,2,3, und entsprechend wird zum Unterprogramm verzweigt.

OR

Funktion:

"ODER"-Verknüpfung zweier numerischer Ausdrücke.

Syntax:

numerischer Ausdruck 1 **OR** numerischer Ausdruck 2

Bemerkungen:

Das Ergebnis einer "ODER"-Verknüpfung ist wahr ("1"), wenn der Ausdruck 1 oder der Ausdruck 2 oder beide wahr sind, sonst ist es falsch ("0").

Wahrheitstabelle:

A	B	A OR B
0	0	0
1	0	1
0	1	1
1	1	1

Bei der "ODER"-Verknüpfung werden die Ausdrücke als 16stellige Binärzahlen interpretiert. Die Ausdrücke müssen dabei im Bereich zwischen -32768 und +32767 liegen. Liegt einer der Ausdrücke nicht in diesem Bereich, erfolgt die Fehlermeldung ERROR 3.
Das Ergebnis einer "ODER"-Verknüpfung weist in jeder binären Stelle eine "1" auf, in der im Ausdruck 1 oder im Ausdruck 2 eine "1" steht, sonst eine "0".

Beispiel:

```
17 OR 3

Ergebnis          16-Bit-Darstellung
19                00000000 00010001   (17)
                  00000000 00000011   (3)
                  00000000 00010011   (17 OR 3 = 19)
```

PASS

Funktion:

Schützt ein Programm vor dem Zugriff durch unbefugte Personen.

Syntax:

PASS Textkonstante

Bemerkungen:

Zum Schutz der Programme vor unberechtigtem Auflisten oder Verändern können die Programme mit einem PASS-Wort versehen werden.
Bei der Verwendung eines PASS-Wortes werden die LIST- und die LLIST-Anweisung nicht ausgeführt. Die Editierfunktionen werden abgeschaltet.
Es können keine Programmzeilen ergänzt oder gelöscht werden.

Das PASS-Wort kann maximal aus sieben Zeichen bestehen. Wird ein längeres PASS-Wort eingegeben, so werden die überzähligen Zeichen ignoriert.

Das PASS-Wort schützt den gesamten Programmspeicher, d.h. sind mehrere Programme gespeichert, läßt sich nicht jedem Programm ein PASS-Wort zugeordnen.

Wird nach der Eingabe eines Programms ein PASS-Wort eingegeben, so ist der Programmspeicher für weitere Programmeingaben gesperrt.

Das PASS-Wort wird durch nochmalige Eingabe wieder aufgehoben.

Ein PASS-Wort kann nicht überschrieben werden, d.h., ist bereits ein PASS-Wort definiert, so führt die Eingabe eines weiteren PASS-Wortes zu einer Fehlermeldung (ERROR 9).

Folgende Punkte sind bei der Verwendung eines PASS-Wortes zu beachten:

1. Das PASS-Wort kann nur direkt im RUN- oder im PRO-Mode eingegeben werden, darf also nicht in einer Programmzeile stehen.
2. Ein PASS-Wort wird gelöscht, indem es nochmals eingegeben wird.
3. Ein durch ein PASS-Wort geschütztes Programm kann mit der NEW-Anweisung gelöscht werden.

4. Ist ein Programm durch ein PASS-Wort geschützt, so kann es nicht auf Band gespeichert werden.
5. Ist ein Programm nicht geschützt, so kann es beim Abspeichern auf Band geschützt werden (siehe CSAVE).
6. Beim Laden von geschützten Programmen ergeben sich folgende Abhängigkeiten:

Operation	Hauptspeicherinhalt
0 CLOAD A	A mit PASS-Wort von A
0 CLOAD B	B ohne PASS-Wort
1 CLOAD A	A mit PASS-Wort von A
1 CLOAD B	B ohne PASS-Wort
2 CLOAD A	A mit PASS-Wort von A
2 CLOAD B	B ohne PASS-Wort

Es bedeuten - 0 = kein Programm im Computer
1 = geschütztes Programm im Computer
2 = ungeschütztes Programm im Computer
A = geschütztes Programm auf Kassette
B = ungeschütztes Programm auf Kassette

Beispiel:

PASS"SHARP"

Das Programm im Hauptspeicher wird durch das PASS-Wort "SHARP" geschützt.

Eingabe:

LIST

Das Programm wird durch das PASS-Wort geschützt und kann nicht gelistet werden.

Nochmalige Eingabe von:

PASS"SHARP"

hebt den Programmschutz wieder auf.

PAUSE

Funktion:

Ausgabe von numerischen Werten und/oder Textausdrücken auf dem Display und automatische Programmfortsetzung.

Syntax:

PAUSE numerischer Ausdruck,(;) numerischer Ausdruck
PAUSE numerischer Ausdruck,(;) Textausdruck
PAUSE Textausdruck,(;) Textausdruck
PAUSE Textausdruck,(;) numerischer Ausdruck

Bemerkungen:

Die Syntax der PAUSE-Anweisung ist identisch mit der der PRINT-Anweisung. Die Bedeutung der einzelnen Sprachelemente (, ;) ist dieselbe. Zum Unterschied zur PRINT-Anweisung wird der Programmablauf nach ca. 0,85 Sekunden fortgesetzt.

Beispiel:

```
10 A$ = "SHARP"
20 B$ = "2000"
30 C$ = "HAMBURG"
40 PAUSE A$
50 PAUSE B$,C$
60 END
```

Augabe nach RUN:

SHARP

und nach ca. 0,85 Sek.

2000 HAMBURG

PI

Funktion:

Die Konstante Pi (π) ist als 3.141592654 gespeichert.

Syntax:

PI oder [SHIFT] [π Exp]

Bemerkungen:

Die Konstante PI kann über die Funktionstasten aufgerufen werden. Dabei wird sie am Display nicht als PI, sondern als π dargestellt.

Beispiel:

```
10 INPUT"RADIUS:";R
20 U=PI*R
30 PRINT"UMFANG=";U
40 END
```

Ausgabe nach RUN:

```
                    Eingabe
RADIUS:_            2.5 [ENTER]

UMFANG=7.8539816
```

POL

Funktion:

Umwandlung von rechtwinkligen Koordinaten in Polarkoordinaten.

Syntax:

POL (numerischer Ausdruck, numerischer Ausdruck)

Bemerkungen:

Der erste numerische Ausdruck gibt die Entfernung von der y-Achse an, der zweite die Entfernung von der x-Achse. Die berechneten Werte für die Entfernung und den Winkel in den Polarkoordinaten werden den Standardvariablen Y (Entfernung) und Z (Winkel) zugeordnet.
Der Wert des Winkels ist dabei von der Winkeleinheit (DEGREE, GRAD, RADIAN) abhängig.
Im BASIC-Programm muß die POL-Funktion einer Variablen zugewiesen werden. Dabei darf die Variable Z nicht verwendet werden. Um Speicherplatz zu sparen, kann die Variable Y als Zuordnungsvariable gewählt werden.

Beispiel:

```
110:  DEGREE
120:  INPUT "STRECKE Y=";
      A
130:  INPUT "STRECKE X=";
      B
140:  Y=POL (A,B)
150:  PRINT "RADIUS R=";Y
160:  PRINT "WINKEL=";Z
170:  END
```

PRINT

Funktion:

Ausgabe von numerischen Werten und Zeichenfolgen am Display.

Syntax:

PRINT numerischer Ausdruck
PRINT Textausdruck
PRINT numerischer Ausdruck ,(;) numerischer Ausdruck
PRINT Textausdruck ,(;) Textausdruck
PRINT numerischer Ausdruck ,(;) Textausdruck
PRINT Textausdruck ,(;) numerischer Ausdruck
PRINT USING Format ; numerischer Ausdruck
PRINT USING Format ; Textausdruck

Bemerkungen:

Der USING-Teil der PRINT-Anweisung spezifiziert das Ausgabeformat, wie unter USING beschrieben.

Mit dem Komma wird die Aufteilung der 16spaltigen Displayzeile in zwei gleich große Felder gesteuert.

Das Semikolon trennt die einzelnen Elemente einer Liste von Ausdrücken und steuert die Position des Cursors.

Wird die Cursor-Positionierung nicht explizit durch die Verwendung eines Semikolons vereinbart, so werden numerische Ausdrücke rechtsbündig und Textausdrücke linksbündig angezeigt.

Nach der Ausführung einer PRINT-Anweisung wird der Programmablauf angehalten, bis man ihn durch Drücken der [ENTER]-Taste wieder startet. Mit der WAIT-Anweisung läßt sich ein Zeitintervall festlegen, nach dessen Ablauf die Programmausführung auch nach einer PRINT-Anweisung automatisch fortgesetzt wird (siehe WAIT).

Die einzelnen Formate der PRINT-Anweisung haben folgende Wirkung:

PRINT numerischer Ausdruck
PRINT Textausdruck

Der in der PRINT-Anweisung spezifizierte Inhalt wird auf der Anzeige ausgegeben; ein Textausdruck wird linksbündig mit maximal 16 Zeichen, ein numerischer Ausdruck rechtsbündig mit maximal 10 Stellen und 2 Exponentialstellen angezeigt.

PRINT numerischer Ausdruck, numerischer Ausdruck (oder Textausdruck)
PRINT Textausdruck, Textausdruck (oder numerischer Ausdruck)

Die Anzeige wird in zwei Felder geteilt. Der erste numerische Ausdruck (bzw. Textausdruck) wird in der linken, der zweite in der rechten Hälfte angezeigt.
Dabei werden Textausdrücke jeweils linksbündig mit maximal 8 Zeichen und numerische Ausdrücke rechtsbündig mit maximal 6 Stellen angezeigt. Bei größeren Zahlen wird automatisch auf die wissenschaftliche Schreibweise umgeschaltet.

PRINT numerischer Ausdruck ; numerischer Ausdruck (oder Textausdruck)
PRINT Textausdruck ; Textausdruck (oder numerischer Ausdruck)

Der Inhalt der PRINT-Liste wird, beginnend mit der jeweiligen Cursor-Position, mit maximal 16 Stellen angezeigt.

Beispiele:

(1)
```
10 A = 3.1415
20 B$ = "SHARP"
30 PRINT A
40 PRINT B$
```

Ausgabe nach RUN:

```
          3.1415
                        [ENTER]
SHARP
```

(2)
```
10 A = 1401
20 B$ = "SHARP"
30 C = 100000 * A
40 PRINT A,B$
50 PRINT B$,A
60 PRINT A,A
70 PRINT C,A
```

Ausgabe nach RUN:

```
   1401.SHARP
                          ENTER
SHARP      1401.
                          ENTER
   1401.   1401.
                          ENTER
 1.401E 08 1401.
```

(3)
```
10 A = 1401
20 B$ = "SHARP"
30 C$ = "PC"
40 D$ = " "
50 E$ = "VON"
60 PRINT E$;D$;B$;D$;C$;A
70 PRINT C$;A;E$;D$;B$
80 PRINT B$;D$;C$;D$;A
90 END
```

Ausgabe nach RUN:

```
VON SHARP PC1401.
                          ENTER
PC1401.VON SHARP
                          ENTER
SHARP PC 1401.
```

(4)
```
5 WAIT 50
10 A = 1.234
20 B$ = "SHARP"
30 C = 5.67
40 D$ = "PC-1401"
50 E$ = "&&###.#"
60 F$ = "###.#&&"
70 PRINT USING E$;B$;A
80 PRINT USING F$;B$;A
90 PRINT USING; B$;A
100 PRINT USING E$;A,C
110 PRINT USING F$;D$
120 PRINT USING "##.####";A,C
130 PRINT USING; A,C
```

Ausgabe nach RUN:

SH 1.2

USING E$ = &&####.# formatiert die Ausgabe mit zwei Textzeichen und zwei Vorkommastellen + eine Nachkommastelle für den numerischen Ausdruck.

SH 1.2

Die Anzeige andert sich nicht. Es ist also gleichgültig, ob zuerst der Textausdruck oder der numerische Ausdruck formatiert wird.

SHARP1.234

Das Format der Anzeige wurde nicht festgelegt. Die Ausgabe der Werte erfolgt daher im Standardformat.

1.2 5.6

USING E$ = &&###.#; die Textformatierung wird bei der Ausgabe von zwei numerischen Werten unterdrückt.

SHPC

USING F$ = ###.#&&; die Formatierung für den numerischen Ausdruck wird bei der Ausgabe von zwei Textausdrücken unterdrückt.

1.2340 5.6700

USING ##.####; es werden vier Nachkommastellen angezeigt; fehlende Stellen werden mit Null (0) aufgefüllt.

1.234 5.67

USING ohne Formatangaben; die Anzeige erfolgt im Standardformat.

Hinweise:

1. Die Anzeige wird nach Drücken der [C-CE] -Taste bzw. nach Fortsetzung des Programms gelöscht.

2. Mit der WAIT-Anweisung kann ein Zeitintervall definiert werden, nach dessen Ablauf die Programmausführung automatisch fortgesetzt wird.

3. Werden mehr Zeichen ausgegeben, als die Anzeige Schreibpositionen hat, so werden nur die ersten 16 Zeichen angezeigt.

4. Wird die PRINT USING-Anweisung verwendet, so bleibt das festgelegte Format so lange erhalten, bis eine weitere USING-Anweisung folgt.

5. Das USING-Format kann durch die RUN-Anweisung oder durch Drücken der Tasten [SHIFT] und [CA C-CE] gelöscht werden.

6. Ist die Option CE-126P an den Rechner angeschlossen, so kann die PRINT-Anweisung auch zur Ausgabe am Drucker verwendet werden. Mit der Anweisung PRINT = LPRINT erfolgt die Umschaltung der Ausgabe vom Display auf den Drucker. Diese Anweisung kann entweder direkt oder in einer Programmanweisung eingegeben werden.
   Die Anweisung wird gelöscht:
   a) Aus-/Einschalten des Computers
   b) RUN
   c) [SHIFT] [CA C-CE]
   d) (Zeilennummer) PRINT = PRINT

Beispiel:

```
10 PRINT = LPRINT
20 A$ = "PC-1401"
30 B$ = "SHARP"
40 PRINT A$,B$
50 PRINT = PRINT
60 PRINT A$,B$
70 END
```

Ausgabe nach RUN:

auf dem Drucker:

PC-1401 SHARP

auf dem Display:

PC-1401 SHARP

PRINT#

Funktion:

Werte von Variablen auf Band speichern.

Syntax:

```
PRINT# Variable
PRINT# Variable*
PRINT# Feldvariable (*)
PRINT# Variablenliste
PRINT# Textkonstante; Variablenliste
```

Bemerkungen:

Beim **PC-1401/1402** unterscheidet man zwischen Programm- und Datenspeicher. Mit der PRINT#-Anweisung werden die Werte einer Variablen oder einer Gruppe von Variablen auf Band gespeichert.

Die Textkonstante in der PRINT#-Anweisung hat die gleiche Bedeutung wie bei der CSAVE-Anweisung. Sie legt den Namen fest, mit dem die Daten auf Band gespeichert werden. Der Name kann aus bis zu 7 Zeichen bestehen.

Die Speicherung kann nur im RUN-Mode sowohl manuell als auch programmgesteuert erfolgen.

Der Rechner kann Programme von Daten unterscheiden. Es ist daher möglich, die zu einem Programm gehörenden Daten mit dem gleichen Namen abzuspeichern.

Geben Sie einen einzelnen Variablennamen oder eine Folge von Variablennamen an, so werden die Inhalte der Variablen auf Band gespeichert.

Die einzelnen Formate der PRINT#-Anweisung haben folgende Wirkung:

PRINT# VARIABLE

Der Wert der definierten Standardvariablen wird abgespeichert

```
z. B. PRINT# A
      PRINT# K
      PRINT# Z1$
```

PRINT# VARIABLE*

Die Werte der Standardvariablen, beginnend mit der definierten Standardvariablen, werden abgespeichert. Sind A-Vektor-Elemente definiert, so werden auch diese abgespeichert.

z. B. PRINT A*
Es werden alle Standardvariablen von A - Z abgespeichert

PRINT K*
Es werden die Standardvariablen von K - Z abgespeichert

PRINT# FELDVARIABLE (*)
Alle Elemente der definierten Feldvariablen werden abgespeichert. Einzelne Elemente können nicht auf Band gespeichert werden.

z. B. PRINT B(*)
PRINT AI (*)
PRINT YZ$ (*)

PRINT # VARIABLENLISTE

Die Werte der Variablen werden in der Reihenfolge ihres Aufrufs abgespeichert

z. B. PRINT # A*, B$(*), AZ(*)

PRINT# TEXTKONSTANTE; VARIABLEN (LISTE)

Diese Anweisung verhält sich wie die vorhergenannten, die Textkonstante spezifiziert einen Dateinamen, unter dem die Variablen abgespeichert werden.

z. B. PRINT# "DATEI"; Y*, A5(*)

RANDOM

Funktion:

Setzt einen Startpunkt für den Zufallszahlengenerator.

Syntax:

RANDOM

Bemerkungen:

Die RANDOM-Anweisung muß vor dem ersten Aufruf der RND-Funktion im Programm stehen. Sie bewirkt, daß bei jeder Programmausführung (mit RUN) eine neue Zufallszahlenfolge initiiert wird.

Beispiel:

```
10 RANDOM
20 FOR I = 1 TO 10
30 X=RND(5)
40 PAUSE X
50 NEXT I
60 END
```

RCP

Funktion:

Ergibt den Reziprokwert eines numerischen Ausdrucks.

Syntax:

RCP numerischer Ausdruck

Beispiel:

```
10 A=100
20 B=RCP A
30 PRINT"X=";A;"RCP X=";B
40 END
```

Ausgabe nach RUN:

```
X=100.RCP X=0.01
```

READ

Funktion:

Weist den angegebenen Variablen Werte aus den DATA-Zeilen zu.

Syntax:

READ numerische Variable
READ Textvariable
READ Variablenliste

Bemerkungen:

Mit der ersten READ-Anweisung wird der erste Wert der DATA-Zeilen der ersten nach der READ-Anweisung aufgeführten Variablen zugeordnet. Die zweite READ-Anweisung weist den zweiten Wert der DATA-Zeilen der entsprechenden Variablen zu. Dabei können numerische Ausdrücke nur numerischen Variablen und Textausdrücke nur Textvariablen zugeordnet werden.

Folgt der READ-Anweisung eine Variablenliste (Reihe von numerischen Variablen und/oder Textvariablen), so werden diesen der Reihenfolge nach die Elemente der DATA-Zeilen zugewiesen.

Sind Variablentyp und Ausdruck nicht identisch (z.B. READ A / DATA "SHARP"), erscheint Fehlermeldung 9.

Ist in der DATA-Anweisung ein arithmetischer Ausdruck (z.B. DATA A+B) enthalten, so wird dieser zuerst berechnet und dann der entsprechenden Variablen zugewiesen.

Hinweise:

1. Grundsätzlich müssen in den DATA-Zeilen ausreichend Elemente gespeichert sein, damit allen auf READ folgenden Variablen ein Wert zugewiesen werden kann. Andernfalls erfolgt eine Fehlermeldung (ERROR 9).

2. Bei indizierten Variablen werden die Indizes zum Zeitpunkt der Ausführung der jeweiligen Wertzuordnung berechnet.

3. Bis zur Ausführung der nächsten READ- (oder RESTORE-)Anweisung bleibt der DATA-Zeiger auf dem zuletzt eingelesenen Element.

REC

Funktion:

Umwandlung von Polarkoordinaten in rechtwinklige Koordinaten.

Syntax:

REC (numerischer Ausdruck, numerischer Ausdruck)

Bemerkungen:

Der erste numerische Ausdruck gibt den Radius an, der zweite den Winkel.
Die berechneten Werte für den x- und den y-Wert werden den Standardvariablen Z (x-Wert) und Y (y-Wert) zugeordnet.
Der Wert des eingegebenen Winkels ist von der Winkeleinheit (DEGREE, GRAD, RADIAN) abhängig.
Im BASIC-Programm muß die REC-Funktion einer Variablen zugewiesen werden. Dabei darf die Variable Z nicht verwendet werden. Um Speicherplatz zu sparen, kann die Variable Y als Zuordnungsvariable gewählt werden.

Beispiel:

```
10:  DEGREE
20:  INPUT "RADIUS R= "; A
30:  INPUT "WINKEL= ";B
40:  Y=REC (A,B)
50:  PRINT "X-WERT= ";Z
60:  PRINT "Y-WERT= ";Y
70:  END
```

REM

Funktion:

Definiert eine Kommentarzeile im Programm.

Syntax:

REM Textausdruck < Kommentar >

Bemerkungen:

An jeder beliebigen Stelle des Programms kann zur Erläuterung eine kommentierende Textzeile eingefügt werden. Die Zeile wird bei der Programmausführung übersprungen.

ACHTUNG: Die REM-Anweisung definiert die gesamte restliche Zeile als Kommentar.

Beispiel: 10 X=45 REM INITIALISIERUNG: A = 0

In diesem Beispiel wird die Wertzuweisung, die hinter der REM-Anweisung steht, als Kommentar aufgefaßt und nie ausgeführt.

Beispiele:

```
10 REM ************************
20 REM *                      *
30 REM *       T E X T        *
40 REM *                      *
50 REM ************************
```

```
10 READ P: REM PREIS PER STÜCK
20 READ N: REM ANZAHL
30 C=P*N:REM GESAMTPREIS
40 END
```

RESTORE

Funktion:

Ermöglicht das erneute Lesen der DATA-Anweisungen von Beginn an.

Syntax:

RESTORE
RESTORE numerischer Ausdruck (Zeilennummer)
RESTORE Textausdruck (Markenname)

Bemerkungen:

Die RESTORE-Anweisung setzt den Zeiger für die DATA-Anweisungen zur ersten DATA-Anweisung zurück, so daß die DATA-Anweisungen wieder von Beginn an gelesen werden können.
Der Wert des numerischen Ausdrucks gibt die Zeilennummer, der Textausdruck den Markennamen der READ-Anweisung an.

Beispiel:

```
10 READ A,B,C
20 RESTORE
30 READ D,E,F,G,H,I
40 FOR Z = 1 TO 9
50 PAUSE A(Z)
60 NEXT Z
70 DATA 10,20,30,40,50,60
```

Ausgabe nach RUN:

```
10.
20.
30.
10.
20.
30.
40.
50.
60.
```

RETURN

Funktion:

Beendet einen Unterprogrammaufruf (GOSUB).

Syntax:

RETURN

Bemerkungen:

Die RETURN-Anweisung in einem Unterprogramm (Subroutine) beendet den Unterprogrammaufruf. Die Programmausführung wird mit der Anweisung, die dem Unterprogrammaufruf (GOSUB) folgt, fortgesetzt.
Unterprogramme können an jeder beliebigen Stelle im Programm stehen, es ist aber empfehlenswert, sie deutlich vom Hauptprogramm abzuheben.

Ein Unterprogramm muß unbedingt mit RETURN abgeschlossen werden. Die GOTO-Anweisung ist nicht zulässig.

Eine RETURN-Anweisung ohne GOSUB-Anweisung führt zu der Fehlermeldung ERROR 5.

Beispiel:

```
 5 WAIT 200
10 PRINT "HAUPTPROGRAMM"
20 GOSUB 100
30 PRINT "HAUPTPROGRAMM"
40 END
100 PRINT"UNTERPROGRAMM"
110 A$ = INKEY$
120 IF A$ = "" THEN 110
130 RETURN
```

Ausgabe nach RUN:

```
HAUPTPROGRAMM

UNTERPROGRAMM

                    < Alphataste >

HAUPTPROGRAMM
```

RIGHT$

Funktion:

Ergibt die letzten *n* Zeichen eines Textausdrucks

Syntax:

RIGHT$ (Textausdruck, numerischer Ausdruck)

Bemerkungen:

Der Wert des numerischen Ausdrucks bestimmt die Anzahl der Zeichen, die von rechts dem Textausdruck entnommen werden und die die Ergebniszeichenfolge bilden.

Beispiele:

```
5 DIM S$(0)*13
10 S$(0) = "SHARP HAMBURG"
20 T$ = RIGHT$ (S$(0), 4)
30 PRINT T$
40 END
```

Ausgabe nach RUN:

```
BURG
```

In Programmzeile 20 werden dem Textausdruck S$(0) die letzten 4 Zeichen (BURG) entnommen und der Variablen T$ zugewiesen.

```
 5 DIM P$(1)*13
10 P$(0) = "8000 MUENCHEN"
20 P$(1) = RIGHT$ (P$(0), 8)
30 PRINT "ORT:"; P$(1)
40 END
```

Ausgabe nach RUN:

```
ORT:MUENCHEN
```

In Programmzeile 20 werden dem Textausdruck P$(0) die letzten 8 Zeichen (MUENCHEN) entnommen und der Variablen P$(1) zugewiesen.

RND

Funktion:

Erzeugt eine Pseudozufallszahl.

Syntax:

RND numerischer Ausdruck

Bemerkungen:

Die Ganzzahlenstelle des numerischen Ausdrucks bestimmt das Intervall, aus dem die Zufallszahlen gezogen werden, die dann das Ergebnis bilden.
Das Intervall ist wie folgt festgelegt:

```
RND X
      X < 0 —— die gleiche Folge von Zufallszahlen wird bei jedem
               Programmlauf generiert
0 < = X < 1 —— 0 < = RND X < 1
X > = 1     —— 1 < = RND X < = X (wenn X Ganzzahl)
               1 < = RND X < = (INT X) + 1 (wenn X Dezimalbruch)
```

Die Genauigkeit der Zufallszahl beträgt 10 Stellen.

Die mit der RND-Anweisung erhaltenen Zahlen sind nicht echt zufällig, da sie aufgrund eines festgelegten Rechenganges ermittelt werden. Sie erscheinen jedoch als zufällig und haben auch dieselben statistischen Eigenschaften wie echte Zufallszahlen.

Nach dem Einschalten erzeugt der **PC-1401/1402** immer dieselbe Folge von Zufallszahlen.

Um eine neue Zufallsreihenfolge zu initiieren, muß die RANDOM-Anweisung zusätzlich zur RND-Anweisung eingegeben werden.

Beispiel:

```
10 WAIT 200
20 PRINT"ZUFALLSZAHL:";RND 6
30 GOTO 20
```

Ausgabe nach RUN:

```
ZUFALLSZAHL:X
```

nach ca. 2 Sek wird die nächste Zufallszahl angezeigt.

ROT

Funktion:

Berechnet die x-te Wurzel eines numerischen Ausdrucks.

Syntax:

numerischer Ausdruck **ROT** numerischer Ausdruck

Bemerkungen:

Der erste numerische Ausdruck gibt den Wurzelexponenten an, der zweite den Radikanden (= Wert unter der Wurzel).

Der Radikand darf nur bei ungeraden Wurzelexponenten negativ sein.

Beispiel:

```
10 A = 100
20 B = 4 ROT A
30 PRINT"4-TE√ ";A;"=";B
40 END
```

Ausgabe nach RUN:

```
4-TE√ 100.=1.0139
```

RUN/GOTO/Definable Keys

Funktion:

Starten der Programmausführung

Syntax:

RUN
RUN numerischer Ausdruck
RUN Textausdruck
GOTO numerischer Ausdruck
GOTO Textausdruck

Bemerkungen:

Nach der Eingabe der RUN-Anweisung und Drücken der [ENTER] Taste wird das Programm mit der niedrigsten Zeilennummer gestartet. Ist eine Zeilennummer (numerischer Ausdruck) oder ein Markenname (Textausdruck) explizit angegeben, wird die Programmausführung mit der spezifizierten Zeile gestartet.

Die RUN-Anweisung kann nur im RUN-Mode eingegeben werden. Ansonsten wird die Fehlermeldung ERROR 9 angezeigt.

Eine weitere Möglichkeit, ein Programm zu starten, bietet die GOTO-Anweisung. Sie ist nur in Verbindung mit einer Zeilennummer (numerischer Ausdruck) oder einem Markennamen (Textausdruck) zulässig.

Die dritte Möglichkeit, ein Programm zu starten, sind die 'Definable Keys' (definierbare Tasten).
Dazu muß die Taste [DEF] und eine der folgenden Tasten gedrückt werden:

[A] [S] [D] [F] [G] [H] [J] [K] [L] [Z] [X] [C] [V] [B] [N] [M] [.] [SPC]

Die Programmausführung beginnt dann mit der Zeile, die das entsprechende Zeichen als Marke (z.B. "A") enthält. Wird eine im Programm nicht enthaltene Marke angegeben, führt dies zur Fehlermeldung ERROR 9.
Wartet der Rechner nach einer INPUT-Anweisung auf eine Eingabe, so kann durch Drücken der Taste [DEF] und einer der 'Definable Keys' zu einer anderen Programmzeile verzweigt werden. Die INPUT-Anweisung wird dann nicht ausgeführt.

Unterschiede zwischen RUN, GOTO und den 'Definable Keys'

Beim Start des Programms werden bestimmte Grundzustände des Rechners gesetzt. Die drei Anweisungen zum Starten eines Programms unterscheiden sich hierbei, wie aus der folgenden Tabelle ersichtlich:

Zustandsänderung	RUN	GOTO	Def. Keys
Die Anzeige wird gelöscht	ja	ja	ja
Die Feldvariablen werden gelöscht	ja	nein	nein
Die RESTORE-Anweisung wird ausgeführt	ja	nein	nein
Das USING-Format wird gelöscht	ja	nein	nein
Das WAIT-Intervall bleibt erhalten	ja	nein	nein
PRINT = LPRINT wird gelöscht	ja	nein	nein

Bei allen drei Startbefehlen wird:

- die FOR-NEXT-Information gelöscht
- die GOSUB-Rücksprungadresse gelöscht
- der Standardvariablen-Speicher **nicht** gelöscht
- der TRACE-Zustand nicht geändert.

Beispiele:

```
RUN

RUN 100

RUN "PROG.1"

GOTO 100

GOTO "PROG.1"
```

DEF A

SGN

Funktion:

Ermittelt das Vorzeichen eines numerischen Ausdrucks.

Syntax:

SGN numerischer Ausdruck

Bemerkungen:

```
SGN X =  1 wenn X > 0
SGN X =  0 wenn X = 0
SGN X = -1 wenn X < 0
```

Beispiel:

```
10 FOR A = 15 TO -15 STEP -15
20 B = SGN A
30 PRINT A,B
40 NEXT A
```

Ausgabe nach RUN:

```
     15.      1.          ENTER
      0.      0.          ENTER
    -15.     -1.
```

SIN

Funktion:

Berechnet den Sinus eines numerischen Ausdrucks in der angegebenen Winkeleinheit.

Syntax:

SIN numerischer Ausdruck

Bemerkungen:

Die Berechnung der Sinusfunktion kann in folgenden drei Winkeleinheiten erfolgen:
DEGREE (0 bis 90°)
GRAD (0 bis 100ᵍ)
RADIAN (0 bis PI/2)
Die gewählte Winkeleinheit wird am Display angezeigt.

Beispiel:

```
200:  DEGREE
210:  INPUT X
220:  LET Y=SIN X
225:  A$="#.##^"
230:  PRINT "SIN" ;X;"=";
      USING A$;Y;USING
```

Ausgabe nach RUN:

```
?            45 [ENTER]

SIN45. = 7.07E-01
```

SQR

Funktion:

Berechnet die Quadratwurzel eines numerischen Ausdrucks.

Syntax:

SQR numerischer Ausdruck

Bemerkungen:

Der Radikand (numerischer Ausdruck unter der Wurzel) darf nicht negativ sein.

Beispiel:

```
10 A = SQR 2
20 B = SQR A
30 PRINT A,B
40 END
```

Ausgabe nach RUN:

```
1.41421 1.18920
```

SQU

Funktion:

Berechnet das Quadrat eines numerischen Ausdrucks.

Syntax:

SQU numerischer Ausdruck

Beispiele:

(1)
```
10 A = SQU 3
20 PRINT A
30 END
```

Ausgabe nach RUN:

```
 9.
```

(2)
```
10 X = 25
20 Y = SQU X
30 PRINT X,Y
40 END
```

Ausgabe nach RUN:

```
 25.    625.
```

STOP

Funktion:

Unterbricht die Programmausführung.

Syntax:

STOP

Bemerkungen:

Durch die STOP-Anweisung wird die Programmausführung unterbrochen. Auf der Anzeige erscheint:

BREAK IN XXX (XXX ... aktuelle Zeilennummer)

Alle Zustände des Rechners (FOR...NEXT, GOSUB, DIM usw.) bleiben erhalten. Dadurch ist es möglich, einen Programmablauf zu unterbrechen (STOP oder BREAK), um die aktuellen Werte von Variablen zu überprüfen und gegebenenfalls zu verändern.

Ein Programm kann mehrere STOP-Anweisungen enthalten. Die Programmausführung kann mit der CONT-Anweisung fortgesetzt werden.

Bei längeren Programmen ist es ratsam, die STOP-Anweisung an das Ende von logischen Programmblöcken zu setzen, so daß die Suche nach eventuellen Programmfehlern erheblich erleichtert wird.

Durch Drücken der [↑]-Taste wird die zuletzt ausgeführte Programmzeile zur Anzeige gebracht.

Durch Betätigen der [↓]-Taste wird die Programmausführung zeilenweise fortgesetzt, wobei als erstes die angezeigte Zeile abgearbeitet und dann die Zeilennummer angezeigt wird.

Betätigt man erneut die [↑]-Taste, so wird wieder die zuletzt ausgeführte Programmzeile angezeigt.

Beispiel:

```
10 PRINT 10
20 STOP
30 PRINT 30
40 PRINT 40
50 PRINT 50
60 PRINT 60
70 PRINT 70
```

Ausgabe nach RUN:

```
                10. ENTER
BREAK IN 20
                    ↑
20: STOP
                    ↓
                30.
                    ↑
30: PRINT.30
                    ↓
30:
  ⋮
                    ↑
70: PRINT 70
```

STR$

Funktion:

Umwandlung des Werts eines numerischen Ausdrucks in einen Textausdruck.

Syntax:

STR$ numerischer Ausdruck

Bemerkungen:

Die Zeichenfolge, die bei der Ausgabe des numerischen Ausdrucks durch eine PRINT-Anweisung im Standardformat auf der Anzeige erscheint, ist das Ergebnis der Textfunktion STR$.

Beispiel:

```
10 S$ = STR$ 7
20 PRINT S$, SQR 7
```

Ausgabe nach RUN:

```
7         2.64575
```

TAN

Funktion:

Berechnet den Tangens eines numerischen Wertes in der angegebenen Winkeleinheit.

Syntax:

TAN numerischer Ausdruck

Bemerkungen:

Die Berechnung der Tangensfunktion kann in folgenden drei Winkeleinheiten erfolgen:
DEGREE (0 bis 90°)
GRAD (0 bis 100ᵍ)
RADIAN (0 bis PI/2)
Die gewählte Winkeleinheit wird am Display angezeigt.

Beispiel:

```
10 DEGREE : INPUT X
20 LET Y = TAN X
30 USING "##.###"
40 PRINT"TAN X=";Y
```

Ausgabe nach RUN:

```
?                    1 [ENTER]

TAN X= 0.017
```

TEN

Funktion:

Exponentialfunktion 10^X.

Syntax:

TEN numerischer Ausdruck

Beispiel:

```
10 A = LOG 100
20 B = TEN A
30 PRINT B
40 END
```

Ausgabe nach RUN:

```
 100.
```

TRON

Funktion:

Einschalten des TRACE-Betriebs.

Syntax:

TRON

Bemerkungen:

Nach Eingabe der TRON-Anweisung wird das Programm wie üblich mit RUN gestartet. Die Programmausführung wird aber nach jeder Zeile automatisch gestoppt, und die entsprechende Zeilennummer wird angezeigt.

Mit der ↑ -Taste kann der jeweilige Zeileninhalt zur Anzeige gebracht, mit der ↓ -Taste die Programmausführung fortgesetzt werden.

Hält man die ↓ -Taste fest, so erfolgt die Programmausführung schnell hintereinander.

Die TRON- und TROFF-Anweisungen können auch im Programm enthalten sein, wodurch begrenzte Programmteile überprüft werden können.

Der TRACE-Betrieb kann durch Ausschalten des Computers, durch die TROFF-Anweisung oder durch Drücken der Tasten SHIFT und CA C·CE wieder ausgeschaltet werden.

Beispiel:

```
10 TRON
20 FOR I = 1 TO 3
30 NEXT I
40 TROFF
```

TROFF

Funktion:

Ausschalten des TRACE-Betriebs.

Syntax:

TROFF

Bemerkungen:

Der TRACE-Betrieb (Programm-Ablaufverfolgung) wird ausgeschaltet.

Beispiel:

```
10 TRON
20 FOR I=1 TO 3
30 NEXT I
40 TROFF
```

USING

Funktion:

Legt das Ausgabeformat von Textausdrücken und Werten numerischer Ausdrücke im Anschluß an eine PRINT-, LPRINT-, oder PAUSE-Anweisung fest.

Syntax:

USING ;
USING Format;

Bemerkungen:

Der **PC-1401/1402** wählt entsprechend dem auszugebenden Wert automatisch ein Ausgabeformat.

Mit der USING-Anweisung kann diese automatische Auswahl abgeschaltet werden, und Sie können selbst bestimmen, in welchem Format die Ausgabe erfolgen soll.

Die USING-Anweisung ist für alle PRINT-, LPRINT- und PAUSE-Anweisungen wirksam.

Das Format kann als Textkonstante oder als Textvariable eingegeben werden. Es bleibt für alle nachfolgenden Ausgabe-Anweisungen erhalten, bis ein neues Format festgelegt wird.

Wird die USING-Anweisung ohne Format eingegeben, so wird die letzte Formatspezifizierung unwirksam und die automatische Formatierung wieder eingeschaltet.

Dasselbe bewirkt die Ausführung der RUN-Anweisung oder das Drücken der Tasten [SHIFT] und [CA / C·CE] .

Zur Festlegung des Formats dienen folgende Sonderzeichen:

(#)... bestimmt die Anzahl der Stellen im numerischen Ausgabefeld.

(.)... legt die Stelle des Dezimalpunktes fest. Die Folge der #-Zeichen nach dem Dezimalpunkt legt die Anzahl der Nachkommastellen fest. In den Nachkommastellen werden fehlende Stellen immer mit Null aufgefüllt. Überzählige Stellen des auszugebenden Wertes werden ignoriert.

(^)... bestimmt die Ausgabe im wissenschaftlichen Format. Dabei können maximal neun Nachkommastellen bestimmt werden.

(&)... bestimmt die Anzahl der Zeichenstellen von Texten. Ist das spezifizierte Format zu klein, so werden nur die formatierten Stellen des Textausdrucks angezeigt. Ist das spezifizierte Format zu groß, so werden die überzähligen Stellen mit Leerstellen aufgefüllt.

Im folgenden sind die unterschiedlichen Formatfestlegungen der USING-Anweisung aufgeführt:

(1) "##...#"

- Anzahl der Ganzzahlenstellen
- Vorzeichenstelle (wird nur bei negativen Zahlen angezeigt)

(2) "##...#."

- Dezimalpunktausgabe
- Anzahl der Vorkommastellen
- Vorzeichenstelle

(3) "##...#.#..#"

- Anzahl der Nachkommastellen
- Dezimalpunkt
- Anzahl der Vorkommastellen
- Vorzeichenstelle

(4) "##.#...#^"

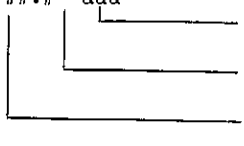
- wissenschaftliches Format
- Mantisse
- Vorzeichenstelle

(5) "&&...&&"

- Anzahl der Textstellen

(6) Kombinierte Formatierung

a) "##.#^&&&"



b) "&&&&##.##"

Hinweise:

1. Reicht das mit der USING-Anweisung festgelegte Ausgabeformat für die Darstellung einer Ganzzahl nicht aus, so wird die Fehlermeldung ERROR 7 angezeigt.

2. Fehlerhafte Formate werden meist erst bei den Ausgabeanweisungen (PRINT, LPRINT oder PAUSE) erkannt und angezeigt.

Beispiele:

(1)
```
10 A = 123.456789
20 USING "####.#"
30 PRINT A
40 USING
50 PRINT A
```

Ausgabe nach RUN:

```
          123.4
```
ENTER
```
      123.456789
```

In der Programmzeile 40 wird auf automatische Formatierung umgeschaltet.

VAL

Funktion:

Berechnet den Wert einer als Zeichenfolge angegebenen Zahl.

Syntax:

VAL Textausdruck

Bemerkungen:

Die Konvertierung des Textausdrucks beginnt mit dem ersten Zeichen des Textes und wird abgebrochen, wenn ein Zeichen vorkommt, das nicht in der Liste (0 . . . 9, . und E) enthalten ist. Die Zeichen + und - sind nur als Vorzeichen zulässig, Zwischenräume werden überlesen.

Beispiele:

(1)
```
10 A = VAL "12.3E12"
20 PRINT A
30 END
```

Ausgabe nach RUN:

```
        1.23E 13
```

(2)
```
10 A = VAL "1 2 3"
20 PRINT A
30 END
```

Ausgabe nach RUN:

```
             123.
```

(3)
```
10 A$ = "4 7 1 1"
20 B = VAL A$
30 PRINT B
40 END
```

Ausgabe nach RUN:

```
            4711.
```

WAIT

Funktion:

Legt das Zeitintervall fest, das vergehen soll, bevor nach einer PRINT-Anweisung der Programmablauf wieder startet.

Syntax:

WAIT num. Ausdruck < Zeitintervall >

Bemerkungen:

Normalerweise wird der Programmablauf nach einer PRINT-Anweisung mit der Eingabe von ENTER fortgesetzt. Mit der WAIT-Anweisung kann ein Zeitintervall festgelegt werden, nach dessen Ablauf die Ausführung automatisch wieder gestartet wird.

Der numerische Ausdruck legt das Intervall fest. Dabei gilt:

1 Einheit = 1/59 Sekunde.

Der Wert des numerischen Ausdrucks darf im Bereich zwischen 0 und 65535 liegen. Damit ergeben sich Zeitintervalle im Bereich von 0 Sekunden bis zu ca. 19 Minuten (= 65535 * 1/59 = 1110,8 Sekunden).

Die WAIT-Anweisung gilt für alle folgenden PRINT-Anweisungen.

Folgt der WAIT-Anweisung kein numerischer Ausdruck, so wird die Automatik abgeschaltet, und die Ausführung muß wieder mit fortgesetzt werden.

Beispiel:

```
10 WAIT 590
20 PRINT"SHARP"
30 PRINT"HAMBURG"
40 END
```

Ausgabe nach RUN:

SHARP

und nach 10 Sekunden:

HAMBURG

ANHANG

ANHANG A TASTENFUNKTIONEN DES PC-1401/1402

ARITHMETISCHE FUNKTIONEN

Funktionsname		Funktion
ABS		Absolutwert
ACS	SHIFT cos⁻¹	Arcuscosinus
AHC	SHIFT arccos cos⁻¹	Areacosinus
AHS	SHIFT arcsin sin⁻¹	Areasinus
AHT	SHIFT arctan tan⁻¹	Areatangens
ASN	SHIFT sin⁻¹	Arcussinus
ATN	SHIFT tan⁻¹	Arcustangens
COS	cos	Cosinus
CUR	SHIFT ∛	Kubikwurzel
DEG	→DEG	Sexagesimalumrechnung
DEGREE		Winkeleinheit Grad
DMS	SHIFT →DMS	Sexagesimalumrechnung
EXP	SHIFT eˣ	Exponentialumrechnung
FAC	SHIFT x!	Fakultät
GRAD		Winkeleinheit Neugrad
HCN	hyp cos	Hyperbelcosinus
HEX/DEC		Sedezimalumrechnung
HSN	hyp sin	Hyperbelsinus
HTN	hyp tan	Hyperbeltangens
INT		Ganzzahlfunktion
LN	ln	Natürlicher Logarithmus
LOG	log	Dekadischer Logarithmus
MEM		Freier Speicherplatz
PI	SHIFT π	Kreiskonstante PI
POL	SHIFT →r∠θ	Polarkoordinatenumrechnung
RADIAN		Winkeleinheit
RANDOM		Zufallsgeneratoranfangswert
RCP	1/x	Reziprokwert
REC	SHIFT →xy	Polarkoordinatenumrechnung
RND		Zufallszahl
ROT	SHIFT ˣ√y	Wurzelberechnungen (x-te Wurzel)
SGN		Vorzeichen
SIN	sin	Sinus
SQU	x²	Quadratzahl
SQR	√	Quadratwurzel
DEC	SHIFT →DEG	Sedezimalumwandlung
TAN	tan	Tangens
TEN	SHIFT 10ˣ	Exponentiation zur Basis 10

Tastenbetätigung		BASIC-Anweisung
[SHIFT]	A	INPUT
[SHIFT]	S	IF
[SHIFT]	D	THEN
[SHIFT]	F	GOTO
[SHIFT]	G	FOR
[SHIFT]	H	TO
[SHIFT]	J	STEP
[SHIFT]	K	NEXT
[SHIFT]	L	LIST
[SHIFT]	,	RUN
[SHIFT]	Z	PRINT
[SHIFT]	X	USING
[SHIFT]	C	GOSUB
[SHIFT]	V	RETURN
[SHIFT]	B	DIM
[SHIFT]	N	END
[SHIFT]	M	CSAVE
[SHIFT]	SPC	CLOAD

ANHANG B LISTE DER FUNKTIONEN UND ANWEISUNGEN

ARITHMETHISCHE FUNKTIONEN

ABS
ACS
AHC
AHS
AHT
ASN
ATN
COS
CUR
DEGREE/GRAD/RADIAN
DEG
DMS
EXP
FAC
HCN
HSN
HTN
INT
LN
LOG
MEM
PI
POL
RANDOM
RCP
REC
ROT
RND
SGN
SIN
SQR
SQU
TEN
TAN

LOGISCHE FUNKTIONEN

AND
NOT
OR

TEXTFUNKTIONEN

ASC
CHR$
INKEY$
LEFT$
LEN
MID$
RIGHT$
STR$
VAL

PROGRAMM-ANWEISUNGEN

AREAD
CLEAR
CONT
DIM
Definable Keys
END
FOR...TO...STEP/NEXT
GOSUB
GOTO
IF...THEN
LET
NEW
ON...GOTO
ON...GOSUB
REM
RETURN
RUN/GOTO
STOP
TRON
TROFF

EINGABE-/AUSGABE-ANWEISUNGEN

BEEP
DATA
INPUT
LIST
PAUSE
PRINT
READ
RESTORE
USING
WAIT

EIN-/AUSGABEANWEISUNGEN MIT DER OPTION CE-126P

INPUT#
PRINT#

DRUCKERFUNKTIONEN (OPTION CE-126P)

LLIST
LPRINT

PROGRAMMSPEICHERUNG AUF MAGNETBAND (OPTION CE-126P)

CLOAD
CLOAD?
CSAVE

PROGRAMMSCHUTZ

PASS

ANHANG C BASIC BEFEHLSOBERSICHT (Kurzfassung)

ARITHMETHISCHE FUNKTIONEN

Funktion	Wirkung	Beispiele
ABS (Ausdruck)	Ergibt den Absolutbetrag eines numerischen Ausdrucks.	ABS(L*0.7) ABS(SIN(X))
ATN (Ausdruck)	Ergibt den Arcustangens eines numerischen Ausdrucks.	ATN(3.5) ATN(A*0.8)
DEGREE/GRAD/RADIAN	Es wird die Winkeleinheit festgelegt, mit der die trigonometrischen Funktionen verarbeitet werden. Die gewählte Winkeleinheit wird angezeigt.	DEGREE RADIAN GRAD
DMS (Ausdruck)	Ermöglicht die Umrechnung vom Dezimalsystem ins Sexagesimalsystem.	DMS 15.2445
DEG (Ausdruck)	Ermöglicht die Umrechnung vom Sexagesimalsystem ins Dezimalsystem.	DEG 15.4125
SIN (Ausdruck)	Ergibt den Sinus eines numerischen Ausdrucks.	SIN 0.755 SIN (A/B)
COS (Ausdruck)	Ergibt den Cosinus eines numerischen Ausdrucks.	COS 0.755 COS (A/B)
TAN (Ausdruck)	Ergibt den Tangens eines numerischen Ausdrucks.	TAN X TAN (X*0.145)
ACS (Ausdruck)	Ergibt den Arcuscosinus eines numerischen Ausdrucks.	ACS 0.8 ACS (A*B)
ASN (Ausdruck)	Ergibt den Arcussinus eines numerischen Ausdrucks.	ASN 0.8 ASN (A*B)
ATN (Ausdruck)	Ergibt den Arcustangens eines numerischen Ausdrucks.	ATN 0.8 ATN (A*B)
AHC (Ausdruck)	Ergibt den Areasinus eines numerischen Ausdrucks.	AHC 1.5 AHC (A*B)
AHS (Ausdruck)	Ergibt den Areacosinus eines numerischen Ausdrucks.	AHS 1.7 AHS (A*B)

Funktion	Wirkung	Beispiele
AHT (Ausdruck)	Ergibt den Areatangens eines numerischen Ausdrucks.	AHT 0.7 AHT (A*B)
LN (Ausdruck)	Ergibt den natürlichen Logarithmus (zur Basis e) eines numerischen Ausdrucks (Ausdruck muß positiv sein).	LN 60 LN (A+B)
LOG (Ausdruck)	Ergibt den Zehnerlogarithmus (zur Basis 10) eines numerischen Ausdrucks (Ausdruck muß positiv sein).	LOG 100 LOG (A+B)
INT (Ausdruck)	Ergibt den abgerundeten ganzzahligen Wert eines numerischen Ausdrucks.	INT (A*B*C)
SGN (Ausdruck)	Ergibt -1 für einen negativen Ausdruck; 0 für einen Null-Ausdruck und +1 für einen positiven Ausdruck.	SGN (-12) SGN (A*B-3)
SQR (Ausdruck)	Ergibt die Quadratwurzel des numerischen Ausdrucks. (Ausdruck muß positiv sein)	SQR (A*B-C)
PI	Die Konstante PI ist als 3.141592654 gespeichert.	
RND (Ausdruck)	Ergibt eine Pseudo-Zufallszahl zwischen 1 und der Ganzzahlenstelle des Ausdrucks. Grenzen: 1 Ausdruck 32768	RND 80 RND (X+Y)
RANDOM	Setzt den Zufallsgenerator auf einen neuen Anfangswert.	RANDOM
MEM	Ergibt die Anzahl der ungenutzten Bytes im Hauptspeicher.	MEM
RCP	Ergibt den Reiprokwert eines numerischen Ausdrucks.	RCP 7
POL	Rechnet Polarkoordinaten in rechtwinklige Koordinaten um.	Y=POL(3,5)
REC	Rechnet rechtwinklige Koordinaten in Polarkoordinaten um.	REC (2,45)

LOGISCHE FUNKTIONEN

Funktion	Wirkung	Beispiele
NOT	Die NOT-Funktion negiert 16 duale Stellen eines numerischen Ausdrucks.	NOT X
AND	Ergibt die UND-Verknüpfung von zwei logischen Ausdrücken	A < 20 AND B > 10
OR	Ergibt die ODER-Verknüpfung von zwei logischen Ausdrücken	A < 20 OR B > 10

TEXTFUNKTIONEN

Funktion	Wirkung	Beispiele
ASC Zeichen	Ergibt den ASCII-Kode eines Zeichens.	ASC "A"
CHR$ ASCII-Kode	Ermittelt aus dem ASCII-Code das zugehörige Zeichen.	CHR$ 85 CHR$ A
LEN Textausdruck	Ermittelt die Länge (Anzahl der Zeichen) eines Text-ausdrucks.	LEN A$
STR$ num. Ausdruck	Wandelt den Wert eines numerischen Ausdrucks in die dezimale Zeichenfolge um.	STR$ SQR 7
VAL Textausdruck	Ergibt einen numerischen Wert entsprechend der im Teaxausdruck enthaltenen Zahl.	VAL "1 2 3" VAL A$+B$
LEFT$(Textausdruck,n)	Ergibt die ersten n Zeichen eines Textausdrucks	LEFT$(A$,5) LEFT$(A$,B+C)
RIGHT$(Textausdruck,n)	Ergibt die letzten n Zeichen eines Textausdrucks	RIGHT$(A$,5) RIGHT$(A$,B+C)

Funktion	Wirkung	Beispiele
MID$(Textausdruck,p,n)	Ergibt den mittleren Teil eines Textausdrucks mit der Zeichenzahl n ab dem p-ten Zeichen.	MID$(A$,5,1) MID$(M$+B$,P,4)
INKEY$	Diese Funktion fragt die Tastatur ab und ergibt je nach gedrückter Taste einen Textausdruck von einem Zeichen Länge, bzw. einen leeren Textausdruck, wenn keine Taste gedrückt wurde.	A$ = INKEY$

EINGABE-/AUSGABE-ANWEISUNGEN

Funktion	Wirkung	Beispiele
PRINT Ausdruck	Mit dieser Anweisung werden numerische oder Textausdrücke auf der Anzeige ausgegeben.	PRINT A PRINT A$ PRINT "SHARP"
,	Ober das Komma wird die Aufteilung der 16-spaltigen Anzeige in zwei gleich große Felder gesteuert.	
;	Das Semikolon trennt die einzelnen Elemente einer Liste von Ausdrücken und steuert die Position des Cursors.	
PAUSE Ausdruck	Mit dieser Anweisung werden numerische oder Textausdrücke auf der Anzeige ausgegeben und der Programmablauf automatischs fortgesetzt.	PAUSE A PAUSE A$ PAUSE "SHARP"
INPUT "Meldung";(,) Variable	Bringt die "Meldung" (falls angegeben) zur Anzeige und weist der Variable Werte zu die über die Tastatur eingegeben werden.	INPUT A INPUT C,A$ INPUT "WERT";B INPUT "NAME";B

Funktion	Wirkung	Beispiele
USING Formatzeichen	Diese Anweisung definiert das Ausgabeformat von Texten und Werten numerischer Ausdrücke im Anschluß an eine PRINT- oder PAUSE-Anweisung. Zur Festlegung des Formats dienen die folgenden Zeichen:	
	# bestimmt die Anzahl der Stellen im numerischen Ausgabefeld. Grenzen: 11 Stellen für Ganzahlen (inkl. Vorzeichen) 12 Stellen für Dezimalzahlen	USING"####"
	. legt die Stelle des Dezimalpunktes fest.	USING"###.##"
	bestimmt die Ausgabe im wissenschaftlichen Format. Es sind maximal 9 Nachkommastellen zulässig.	USING"##.# "
	& bestimmt die Anzahl der Zeichenstellen von Texten und Textausdrücken.	USING"&&&&&&&"
	^ bestimmt die Ausgabe nach dem wissenschaftlichen Format. Dabei können maximal neun Nachkommastellen bestimmt werden.	USING"##.#^"
WAIT Zeitintervall	Diese Anweisung legt das Zeitintervall fest, das vergehen soll bevor nach einer PRINT-Anweisung der Programmablauf wieder startet. Hinweis: Die WAIT-Anweisung wirkt auf alle folgenden PRINT-Anweisungen.	WAIT 54.4
BEEP Anzahl der Signaltöne	Mit dieser Anweisung wird der akustische Signalgeber (Pips) angesteuert.	BEEP 10
DATA Konstantenliste	Diese Anweisung stellt Daten zur Eingabe mit der READ-Anweisung zur Verfügung.	DATA 4.15,7.45 DATA "SHARP", "ENDE"
READ Variablenliste	Diese Anweisung weist den angegebnen Variablen Werte aus den DATA-Zeilen zu.	READ A,B,C READ A$,B$,C$

EIN-/AUSGABEANWEISUNGEN MIT DER OPTION CE-126P

Funktion	Wirkung	Beispiele
RESTORE	Setzt den DATA-Zeiger auf das erste Element der ersten DATA-Anweisung zurück.	RESTORE RESTORE 20 RESTORE "xxx"
PRINT# Variable	Mit dieser Anweisung werden die Werte der Variablen auf Band gespeichert.	PRINT# C PRINT# J,B$(*)
INPUT# Variable	Mit dieser Anweisung werden die auf Band gespeicherten Werte gelesen und den Variablen zugeordnet.	INPUT# C INPUT# J,B$(*)
LPRINT Variable	Mit dieser Anweisung werden numerische Werte oder Textausdrucke auf dem Drucker ausgegeben.	LPRINT A LPRINT "SHARP" LPRINT A$;B$;C

PROGRAMM-ANWEISUNGEN

Funktion	Wirkung	Beispiele
CLEAR	Diese Anweisung löscht alle Variablen und Felder im Hauptspeicher und setzt die Standardvariablen auf 0 (Null)	CLEAR
NEW	Diese Anweisung löscht den Hauptspeicher.	NEW
RUN/GOTO	Mit diesen Anweisungen wird die Programmausführung gestartet.	RUN RUN 200 RUN "TEST" GOTO 200 GOTO "TEST"
Definable Keys (Definierbare Tasten)	Eine weitere Möglichkeit, ein Programm zu starten, sind die sogenannten 'Definable Keys'. Dazu muß die DEF -Taste und die dem jeweiligen Programm zugeordnete Taste gedrückt werden.	DEF A

Funktion	Wirkung	Beispiele
DEF (Fortsetzung)	Definierbare Tasten: A,S,D,F, G,H,J,K,L, ',Z,X,C,V,B,N,M, SPC (Leertaste).	
AREAD	Diese Anweisung liest den vor dem Programmstart angezeigten Wert in eine Variable ein.	"S" AREAD X
STOP	Diese Anweisung unterbricht die Programmausführung	STOP
CONT	Diese Anweisung setzt die Programmausführung nach einer Unterbrechung durch eine STOP-Anweisung oder durch BREAK fort.	CONT
END	Diese Anweisung beendet die Programmausführung	END
LIST Zeilennummer "Markenname"	Mit dem Kommando LIST kann jeweils eine Programmzeile zur Anzeige gebracht ('gelistet') werden.	LIST LIST 100 LIST "LABEL 1"
TRON	Trace (Ablaufverfolgung) einschalten. ⇩ -Taste: Zeileninhalt anzeigen ↵ -Taste: Programmausführung fortsetzten	TRON
TROFF	Trace (Ablaufverfolgung) ausschalten.	TROFF
GOTO Zeilennummer "Markenname"	Diese Anweisung verzweigt zur angegebenen Zeilennummer oder zum angegebenen Markennamen	GOTO 100 GOTO "LABEL 1"
ON Ausdruck GOTO Zeile Zeile 2, ..., Zeile Marke 1, Marke 2, .. Marke k	Diese Anweisung verzweigt zur angegeben Zeilennummer oder zum angegeben Markennamen in Abhängigkeit vom Wert eines numerischen Ausdrucks.	ON I GOTO 100, 200,300,... ON I GOTO "A", "B","C",...

Funktion	Wirkung	Beispiele
GOSUB Zeilennummer Markenname	Diese Anweisung verzweigt zu dem Unterprogramm mit der angegeben Zeilennummer oder dem angegebenen Markennamen.	GOSUB 200 GOSUB "LABEL 1"
ON Ausdruck GOSUB Zeile Zeile 2, ..., Zeile Marke 1, Marke 2,... Marke k	Diese Anweisung verzweigt zu dem Unterprogramm mit der angegeben Zeilennummer oder dem angegebenen Markennamen in Abhängigkeit vom Wert eines numerischen Ausdrucks.	ON I GOSUB 10, 20,30,... ON I GOSUB"A", "B","C",...
RETURN	Diese Anweisung beendet ein Unterprogramm. Die Programmausführung springt zur Adresse des letzten Unterprogrammaufrufs bzw. ON...GOSUB-Aufrufs zurück.	RETURN
IF...THEN	Die Anweisung ermöglicht eine Verzeigung des Programms in Abhängigkeit vom Ergebnis eines logischen Vergleichsausdrucks.	IF A = 3 THEN 10 IF A$ = "X" THEN 100
FOR...TO...STEP/NEXT	Diese Anweisung dient zur wiederholten Ausführung des zwischen FOR...TO und NEXT eingeschlossenen Programmteils. Mit Hilfe der STEP-Anweisung kann die Schrittweite des FOR...TO-Ausdrucks bestimmt werden. Entfällt diese, so ist die Schrittweite mit 1 festgelegt.	FOR I=1TO10 NEXT I FOR A = 1TO10 STEP 10 NEXT A
DIM	Mit dieser Anweisung werden Feldvariable (Arrays) explizit dimensioniert.	DIM B(25) DIM B(25),B$(1) DIM Z(10,8) DIM Z$(12,5)
LET	Diese Anweisung weist einer Variablen (einfach oder indiziert) einen Wert zu.	LET A = 7 LET A$="SHARP"

Funktion	Wirkung	Beispiele
REM	Diese Anweisung definiert eine Kommentarzeile im Programm	REM TESTPRO
RANDOM	Diese Anweisung setzt den Pseudozufallsgenerator auf einen neuen Ausgangswert.	RANDOM
LLIST	Mit dieser Anweisung wird das Programm mit dem Drucker (Option CE-126P) gelistet.	LLIST

PROGRAMMSPEICHERUNG AUF MAGNETBAND

Funktion	Wirkung	Beispiele
CSAVE Programmname	Mit dieser Anweisung werden Programme auf Band gespeichert. Soll das Programm als geschütztes Programm gespeichert werden, so muß nach dem Programmnamen noch ein PASS-Wort eingegeben werden.	CSAVE CSAVE "PRG A" CSAVE,"GEHEIM" CSAVE "PRG A", "GEHEIM"
CLOAD Programmname	Mit dieser Anweisung werden Programme vom Band geladen.	CLOAD CLOAD "PRG A"
CLOAD?	Mit dieser Anweisung wird das im Hauptspeicher gespeicherte Programm mit dem am Band gespeicherten Programm verglichen.	CLOAD? CLOAD? "PRG A"

PROGRAMMSCHUTZ

Funktion	Wirkung	Beispiele
PASS "Textausdruck"	Diese Anweisung schützt ein Programm vor dem Zugriff unbefugter Personen. Das PASS-Wort kann durch nochmalige Eingabe wieder gelöscht werden. Ein geschütztes Programm kann nicht auf Band gespeichert werden. Ist ein Programm nicht geschützt, so kann es als geschütztes Programm auf Band gespeichert werden (siehe CSAVE).	PASS "GEHEIM"

ANHANG D STANDARDVARIABLEN IM PC-1401/1402

```
A = A$ = A(1)  = A$(1)
B = B$ = A(2)  = A$(2)
C = C$ = A(3)  = A$(3)
D = D$ = A(4)  = A$(4)
E = E$ = A(5)  = A$(5)
F = F$ = A(6)  = A$(6)
G = G$ = A(7)  = A$(7)
H = H$ = A(8)  = A$(8)
I = I$ = A(9)  = A$(9)
J = J$ = A(10) = A$(10)
K = K$ = A(11) = A$(11)
L = L$ = A(12) = A$(12)
M = M$ = A(13) = A$(13)
N = N$ = A(14) = A$(14)
O = O$ = A(15) = A$(15)
P = P$ = A(16) = A$(16)
Q = Q$ = A(17) = A$(17)
R = R$ = A(18) = A$(18)
S = S$ = A(19) = A$(19)
T = T$ = A(20) = A$(20)
U = U$ = A(21) = A$(21)
V = V$ = A(22) = A$(22)
W = W$ = A(23) = A$(23)
X = X$ = A(24) = A$(24)
Y = Y$ = A(25) = A$(25)
Z = Z$ = A(26) = A$(26)
```

ANHANG E OPERATOREN

E.1 Arithmetische Operatoren

+	Addition
-	Subtraktion
*	Multiplikation
/	Division
^	Exponentation

E.2 Textfunktionsoperatoren

+	Verkettung

E.3 Logische Vergleichsoperatoren

	Bedeutung in numerischen Ausdrücken	Bedeutung in Textausdrücken
<	kleiner als	ASCII-Kode kleiner
>	größer als	ASCII-Kode größer
<=	kleiner gleich	ASCII-Kode kleiner oder gleich
>=	größer gleich	ASCII-Kode größer oder gleich
<>	ungleich	ASCII-Kode ungleich

E.4 Reihenfolge der Operatoren

1. Klammern
2. Abruf von PI, MEM, Variablen
3. Funktionsoperationen in Bezug auf das folgende Argument
4. Exponentation
5. Vorzeichen + , -
6. Multiplikation, Division
7. Addition, Subtraktion.
8. Vergleichsoperationen
9. Logische Operationen

ANHANG F RECHENBEREICH

Funktion	Rechenbereich	Anmerkung
$\sin x$ $\cos x$ $\tan x$	DEG: $\lvert x \rvert < 1 \times 10^{10}$ RAD: $\lvert x \rvert < \frac{\pi}{180} \times 10^{10}$ GRAD: $\lvert x \rvert < \frac{10}{9} \times 10^{10}$ Für tan x gelten folgende Einschränkungen DEG: $\lvert x \rvert = 90\,(2n-1)$ RAD: $\lvert x \rvert = \frac{\pi}{2}\,(2n-1)$ n = Ganze Zahl GRAD: $\lvert x \rvert = 100\,(2n-1)$	
$\sin^{-1} x$ $\cos^{-1} x$	$-1 \leqq x \leqq 1$	
$\tan^{-1} x$	$\lvert x \rvert < 1 \times 10^{100}$	
$\ln x$ $\log x$	$1 \times 10^{-99} \leqq x < 1 \times 10^{100}$	$(\ln x = \log_e x)$
e^x	$-1 \times 10^{100} < x \leqq 230.2585092$	(e = 2.718281828)
10^x	$-1 \times 10^{100} < x < 100$	
y^x	• $y > 0$: $-1 \times 10^{100} < x \log y < 100$ • $y = 0$: $x > 0$ • $y < 0$: x: ganzzahlig oder 1/x: ungerade $-1 \times 10^{100} < x \log \lvert y \rvert < 100$	$y^x = 10^{x \cdot \log y}$
$\sqrt[x]{y}$	• $y > 0$: $-1 \times 10^{100} < \frac{1}{x} \log y < 100,\ x \neq 0$ • $y = 0$: $x > 0$ • $y < 0$: x oder 1/x: ganzzahlig $(x \neq 0)$ $-1 \times 10^{100} < \frac{1}{x} \log \lvert y \rvert < 100$	$\sqrt[x]{y} = 10^{\frac{1}{x} \cdot \log y}$
$\sqrt[3]{x}$	$\lvert x \rvert < 1 \times 10^{100}$	
sinh x cosh x tanh x	$-227.9559242 \leqq x \leqq 230.2585092$	
$\sinh^{-1} x$	$\lvert x \rvert < 1 \times 10^{50}$	
$\cosh^{-1} x$	$1 \leqq x < 1 \times 10^{50}$	
$\tanh^{-1} x$	$\lvert x \rvert < 1$	
$\sqrt{x}$	$0 \leqq x < 1 \times 10^{100}$	
x^2	$\lvert x \rvert < 1 \times 10^{50}$	
$\frac{1}{x}$	$\lvert x \rvert < 1 \times 10^{100}$ $x \neq 0$	
n!	$0 \leqq n \leqq 69$ (n = Ganze Zahl)	
→ DEG	$\lvert x \rvert < 1 \times 10^{100}$	
→ DMS	$\lvert x \rvert < 1 \times 10^{100}$	
HEX → DEC	$0 \leqq x < 2540BE3FF$ $FDABF41C01 \leqq x \leqq FFFFFFFFFF$	x ist in "HEX" eine ganze Zahl
DEC → HEX	$\lvert x \rvert \leqq 9999999999$	x = ganzzahlig

Funktion		Rechenbereich	Anmerkung
$x, y \rightarrow r, \theta$		$(x^2 + y^2) < 1 \times 10^{100}$ $\frac{y}{x} < 1 \times 10^{100}$	$r = \sqrt{x^2 + y^2}$ $\theta = \tan^{-1} \frac{y}{x}$
$r, \theta \rightarrow x, y$		$r < 1 \times 10^{100}$, $\lvert r \sin\theta \rvert < 1 \times 10^{100}$ $\lvert r \cos\theta \rvert < 1 \times 10^{100}$	$x = r \cos\theta$ $y = r \sin\theta$ θ ist in derselben Bedingung wie x von sin x, cos x
Statistische Berechnungen	Data CD	$\lvert x \rvert < 1 \times 10^{50}$ $\lvert y \rvert < 1 \times 10^{50}$ $\lvert \Sigma x \rvert < 1 \times 10^{100}$ $\Sigma x^2 < 1 \times 10^{100}$ $\lvert \Sigma y \rvert < 1 \times 10^{100}$ $\Sigma y^2 < 1 \times 10^{100}$ $\lvert \Sigma xy \rvert < 1 \times 10^{100}$ $\lvert n \rvert < 1 \times 10^{100}$	
Statistische Berechnungen	$\bar{x}$	$n \neq 0$	
	Sx	$n \neq 1$ $0 \leqq \frac{\Sigma x^2 - n\bar{x}^2}{n-1} < 1 \times 10^{100}$	
	σx	$n \neq 0$ $0 \leqq \frac{\Sigma x^2 - n\bar{x}^2}{n} < 1 \times 10^{100}$	
	$\bar{y}$	$n \neq 0$	
	Sy	$n \neq 1$ $0 \leqq \frac{\Sigma y^2 - n\bar{y}^2}{n-1} < 1 \times 10^{100}$	
	σy	$n \neq 0$ $0 \leqq \frac{\Sigma y^2 - n\bar{y}^2}{n} < 1 \times 10^{100}$	
	r	$n \neq 0$ $0 < \lvert (\Sigma x^2 - n\bar{x}^2) \cdot (\Sigma y^2 - n\bar{y}^2) \rvert < 1 \times 10^{100}$ $\left\lvert \Sigma xy - \frac{\Sigma x \cdot \Sigma y}{n} \right\rvert < 1 \times 10^{100}$ $\left\lvert \frac{\Sigma xy - \frac{\Sigma x \cdot \Sigma y}{n}}{\sqrt{(\Sigma x^2 - n\bar{x}^2) \cdot (\Sigma y^2 - n\bar{y}^2)}} \right\rvert < 1 \times 10^{100}$	
	b	$n \neq 0$ $0 < \lvert \Sigma x^2 - n\bar{x}^2 \rvert < 1 \times 10^{100}$ $\left\lvert \Sigma xy - \frac{\Sigma x \cdot \Sigma y}{n} \right\rvert < 1 \times 10^{100}$ $\left\lvert \frac{\Sigma xy - \frac{\Sigma x \cdot \Sigma y}{n}}{\Sigma x^2 - n\bar{x}^2} \right\rvert < 1 \times 10^{100}$	
	a	a ist in derselben Bedingung wie b, und $\lvert \bar{y} - b\bar{x} \rvert < 1 \times 10^{100}$	
	y'	$\lvert a + bx \rvert < 1 \times 10^{100}$	
	x'	$\left\lvert \frac{y-a}{b} \right\rvert < 1 \times 10^{100}$	

ANHANG G ABGELEITETE MATHEMATISCHE FUNKTIONEN

Funktion	
Sekans	SEC(X)=1/COS(X)
Cosekans	CSC(X)=1/SIN(X)
Cotangens	COT(X)=1/TAN(X)
Arcussinus	ARCSIN(X)=ATN(X/SQR(-X*X+1))
Arcuscosinus	ARCCOS(X)=-ATN(X/SQR(-X*X+1))+1.5708
Arcussekans	ARCSEC(X)=ATN(SQR(X*X-1))+(SGN(X)-1)*1.5708
Arcuscosekans	ARCCSC(X)=ATN(1/SQR(X*X-1))+(SGN(X)-1)*1.5708
Arcuscotangens	ARCCOT(X)=-ATN(X)+1.5708
Hyperbelsinus	SINH(X)=(EXP(X)-EXP(-X))/2
Hyperbelcosinus	COSH(X)=(EXP(X)+EXP(-X))/2
Hyperbeltangens	TANH(X)=-EXP(-X)/(EXP(X)+EXP(-X))*2+1
Hyperbelsekans	SECH(X)=2/(EXP(X)+EXP(-X))
Hyperbelcosekans	CSCH(X)=2/(EXP(X)-EXP(-X))
Hyperbelcotangens	COTH(X)=EXP(-X)/(EXP(X)-EXP(-X))*2+1
Hyperbel-Areasinus	ARSINH(X)=LOG(X+SQR(X*X+1))
Hyperbel-Areacosinus	ARCOSH(X)=LOG(X+SQR(X*X-1))
Hyperbel-Areatangens	ARTANH(X)=LOG((1+X)/(1-X))/2
Hyperbel-Areasekans	ARSECH(X)=LOG((SQR(-X*X+1)+1)/X)
Hyperbel-Areacosekans	ARCSCH(X)=LOG((SGN(X)*SQR(X*X+1)+1)/X)
Hyperbel-Areacotangens	ARCOTH(X)=LOG((X+1)/(X-1))/2

ANHANG H PROGRAMM-HAUPTSPEICHER-BEGRENZUNG

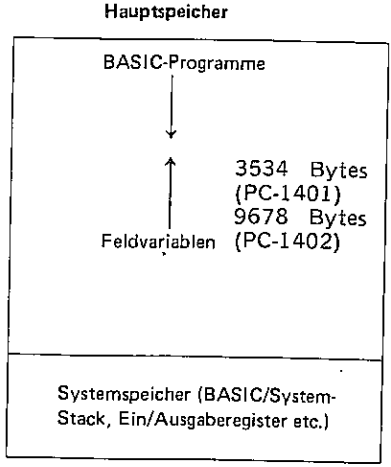
Zulässige Bereiche:

Zeichenketten: bis zu 80 Zeichen

zulässige Zeilennummern: 0 bis 65529 (einschließlich)

Länge der Programmzeile: bis zu 80 Zeichen

Speicherorganisation:



Standardvariablenspeicher (208 Bytes)

A ~ Z	=	A(1)	~	A(26) bzw.
A$ ~ Z$	=	A$(1)	~	A$(26)

ANHANG I FEHLERMELDUNGEN

ERROR Erklärung der Fehlermeldung

1 **Syntax Fehler:**
z.B. 10: PRINTT "SHARP"
Die PRINT-Anweisung wurde falsch eingegeben.

2 **Unzulässige Berechnungen:**
z.B. der Rechenbereich des **PC-1401/1402** wurde überschritten, oder es wurde versucht, eine Division durch 0 einzugeben.

3 **Fehler in der Speicherbelegung:**
Es wurde z. B. versucht, eine Operation auszuführen, bei der ein Konflikt endstanden ist zwischen der Wirkung der Operation und der Speicherorganisation.
z.B. 10 FOR I = 1 TO 35000
Der Wert der Schleifenvariable ist höher als zugelassen.

4 **Fehler in der Zeilennummer:**
Sie haben eine unzulässige Zeilennummer verwendet.

5 **Schachtelungsfehler:**
Zu einer NEXT- bzw. RETURN-Anweisung exestiert keine entsprechende FOR- bzw. GOSUB-Anweisung.
z.B. 10: FOR A = 1 TO 10
20: NEXT B

6 **Speicherüberlauf:**
Die Speicherkapazität des **PC-1401/1402** wurde überschritten.

7 **Formatfehler:**
z.B. 10: USING"####"
20: A=123*20
30: PRINT A

8 **Fehler bezüglich einer Option:**
Die Informationsübertragung zwischen dem **PC-1401/1402** und der Option **CE-126P** funktioniert nicht. Batteriespannung des **CE-126P** überprüfen. Verbindung **PC-1401/1402/CE-126P** überprüfen. Verbindung zum externen Kassettenrekorder überprüfen.

9 **Sonstige:**
Ein anderer Fehler als bisher spezifiziert ist aufgetreten.
z.B. CHR$ (1)
1 ist hier ein unzulässiger Wert
oder
10: A = 5 : PRINT A$
Es kann nur entweder A oder A$ als Variable verwendet werden.

ANHANG J ASCII-CODE TABELLE

Hex / Binary	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	E 1110	F 1111
0 0000	0 NUL	16	32 SPACE	48 0	64 @	80 P	96	112	128	224	240
1 0001	1	17	33 !	49 1	65 A	81 Q	97	113	129	225	241
2 0010	2	18	34 "	50 2	66 B	82 R	98	114	130	226	242
3 0011	3	19	35 #	51 3	67 C	83 S	99	115	131	227	243
4 0100	4	20	36 $	52 4	68 D	84 T	100	116	132	228	244
5 0101	5	21	37 %	53 5	69 E	85 U	101	117	133	229	245
6 0110	6	22	38 &	54 6	70 F	86 V	102	118	134	230	246
7 0111	7	23	39 '	55 7	71 G	87 W	103	119	135	231	247
8 1000	8	24	40 (	56 8	72 H	88 X	104	120	136	232	248
9 1001	9	25	41)	57 9	73 I	89 Y	105	121	137	233	249
A 1010	10	26	42 *	58 :	74 J	90 Z	106	122	138	234	250
B 1011	11	27	43 +	59 ;	75 K	91 [	107	123	139	235	251 π
C 1100	12	28	44 ,	60 <	76 L	92 ¥	108	124	140	236	252 √
D 1101	13	29	45 -	61 =	77 M	93]	109	125	141	237	253
E 1110	14	30	46 .	62 >	78 N	94 ^	110	126	142	238	254
F 1111	15	31	47 /	63 ?	79 O	95 _	111	127	143	239	255

ANHANG K REFERENZLISTE

Gottfried, Byron S.

Programmieren mit BASIC
Düsseldorf: McGraw-Hill
Book Co., 1978
ISBN 0-07-092022-2

Haase, V./W. Stucky

BASIC. Programmieren für Anfänger
Mannheim: Bibliographisches Institut, 1977
BI-Hochschultaschenbuch 744
ISBN 3-411-00744-3

Kahlig, Peter

Graphische Darstellung mit dem Taschencomputer PC-1211 (SHARP)
aus der Reihe
Anwendung programmierbarer Taschenrechner Nr. 14
Braunschweig: Vieweg-Verlag, 1982
ISBN 3-528-04203-6

Kaucher, E. et al.,

Programmiersprachen im Griff
Band 3: BASIC
Mannheim: Bibliographisches Institut, 1981
BI-Hochschultaschenbuch 797
ISBN 3-411-00797-4

Kreth, Horst

Lehr- und Übungsbuch für die Rechner SHARP PC-1210 und PC-1211
aus der Reihe
programmieren von Taschenrechnern Nr. 7
Braunschweig: Vieweg-Verlag, 1982
ISBN 3-528-04212-5

Pol, Bernd

Wie man die BASIC programmiert
Einführung – Techniken – Fallstudien
aus der Reihe
CHIP-Wissen Software
Würzburg: Vogel-Verlag, 1981
ISBN 3-8023-0637-6

ANHANG L TECHNISCHE DATEN PC-1401/1402

Modell:	**PC-1401/1402** Taschencomputer
Rechenstellen:	10 Stellen Mantisse, 2 Stellen Exponent
Rechensystem:	AEL mit Hierarchie
Anzeige:	Flüssigkristallanzeige, 16 Stellen in 5x7-Matrix
Tastatur:	76 Tasten Alphatastatur (Schreibmaschine) Zifferntastatur Funktionstasten
Programmiersprache:	BASIC
CPU:	CMOS 8-Bit
Betriebssystem:	ROM 40-kByte
Speicherkapazität:	RAM: 500 Byte System 3534 Byte BASIC-Programm (PC-1401) 9678 Byte BASIC-Programm (PC-1402) 208 Byte Standardvariablen
Speicherschutz:	Programm-Daten-Speicher sind beim Ausschalten geschützt.
Stromversorgung:	6 Volt, 2 Stück Lithium-Batterien (z.B. VARTA CR2032, UCAR CR2032)
Stromverbrauch:	0,03 W bei 6 Volt
Betriebszeit:	ca. 300 Stunden mit einem Batteriesatz
Betriebstemperatur:	0°C ... 40°C
Abmessungen:	170 x 72 x 9,5 (BxHxT) in mm
Gewicht:	ca. 150g
Zubehör:	1 Tastaturabdeckung, 2 Batterien (eingebaut) 1 Tastaturschablone, 1 Bedienungsanleitung
Option:	CE-126P: Thermodrucker mit integriertem Kassetten-Interface für ein externes Bandgerät

ANHANG M TECHNISCHE DATEN CE-126P

Modell: CE-126 Thermodrucker mit integriertem Kassetten-Interface

1. Drucker

Zeichen/Zeile: 24

Druckgeschwindigkeit: 0,8 Zeilen/sek.

Druckprinzip: Thermodruck in 5x7-Matrix

Papiertransport: manuell oder programmgesteuert

Papierabmessungen: 58 x 18 mm Thermopapier (EA-1250P)

2. Integriertes Kassetten-Interface

An das integrierte Kassetten-Interface kann jeder handelsübliche Kassettenrecorder angeschlossen werden, der folgende Bedingungen erfüllt:

- Fernbedienung (REMOTE), Klinkenbuche 2,5 mm ø
- Bandzählwerk
- Mikrofoneingang (MIC), Klinkenbuchse 3,5 mm ø
- Ausgang (EAR), Klinkenbuchse 3,5 mm ø

Daneben sind folgende technische Daten gefordert:

- Eingangsimpedanz (MIC)	200...1000 Ohm
- Eingangsempfindlichkeit	Kleiner als 3 mV (-50 dB)
- Ausgangsimpedanz (EAR)	Kleiner als 10 Ohm
- Ausgangsleistung (EAR)	Größer als 1 V_{ss}
- Klirrfaktor	Kleiner als 15 % im Bereich zwischen 2...4 kHz
- Gleichlaufschwankungen	0,3 % max. (W.R.M.S)

ANHANG N. INDEX

ABS 102
ACS 103
Addition 50
AND 107
AREAD 108
Anzeige 20
Arithmetische Funktionen 91
ASC 109
ASN 110
ATN 111
Auf-/Abrundung 43
Ausschalten 15
BEEP 112
Betriebsart 21
Betriebshinweise 9
CHR$ 113
CLEAR 114
CLOAD 115
CLOAD?. 116
CONT 117
COS 118
CSAVE 119
CUR 121
DATA 122
Datenspeicherung 28
DEGREE/GRAD/RADIAN 123
DEG 124
DIM 125
DMS 129
Definable Keys 191
Division 50
Drucker 25
END 130
Editierfunktionen 33
Einfache Variable 85
Einschalten 15, 72
EXP 131
Exponentialfunktionen 57
FAC 132
Fakultat 57
Fehlermeldung/Fehlersuche 99
Feldvariable 87
FOR...TO...STEP/NEXT 133
GOSUB 136
GOTO 138

HCS 139
Hexadezimalumwandlung 46
Hexadezimalzahlen 74
HSN 140
HTN 141
Hyperbel- und inverse Hyperbelfunktionen 56
IF...THEN 142
Indizierte Variable 85
INKEY$ 145
INPUT 146
INPUT# 149
INT 150
Kassetten-Interface 27
Koordinatenumwandlung 46
LEFT$ 151
LEN 152
LET 153
LIST 154
LLIST 155
LPRINT 156
LN 159
LOG 160
Logarithmische Funktionen 56
Löschen einer Zeile 98
Löschen der Eingabe oder einer Fehlermeldung 42
MEM 161
MID$ 162
Multiplikation 50
NEW 163
Nachkommastellen 43
NOT 164
Numerische Feldvariable 88
ON...GOTO 165
ON...GOSUB 167
OR 168
PASS 169
PAUSE 171
PI 172
PRINT 174
PRINT# 179
Programmaufbau. 95
Programmausführung. 99
Programmeingabe 96
Programmerstellung. 94
POL 173
Potenzfunktion 51
RANDOM 181
RCP 182
READ 183
RETURN 187

Rechenbereich 64
Rechengenauigkeit 70
REM 185
RESTORE 186
RETURN 187
Reziprok-Rechnung 52
RIGHT$ 188
RND 189
ROT 190
RUN/GOTO 191
Schlüsselwörter, vorprogrammierte 100
SGN 193
SIN 194
SQR 195
SQU 196
Speicherrechnung 52
Statistische Berechnungen 57
STOP 197
STR$ 199
Stromversorgung 9
Subtraktion 50
TAN 200
Tastaturabdeckung 11
Tastenfunktionen 17
Textausdrucke 75
Textfunktionen 92
Textfeldvariable 89
Trigonometrische Funktionen und Umkehrfunktionen 55
TRON 202
TROFF 203
USING 204
Vergleichsausdrücke, logische 81
VAL 207
Variable A 86
Variablen 83
WAIT 208
Winkelumwandlung 45
Wissenschaftliche Screibweise 44, 71
Wurzelfunktion 51

SHARP CORPORATION

OSAKA, JAPAN

1984 © SHARP CORPORATIO
Printed in Japan/Imprimé au Japo
4K2.6T(TINSG4363CCZ